



ESCUELA SUPERIOR DE INGENIERÍA

INGENIERÍA TÉCNICA EN INFORMÁTICA DE GESTIÓN

MissingWords:

Videojuego Android para reforzar el aprendizaje
de vocabulario en idiomas

Adrián Pérez Fernández

2 de diciembre de 2014



ESCUELA SUPERIOR DE INGENIERÍA

INGENIERO TÉCNICO EN INFORMÁTICA DE GESTIÓN

MissingWords:

Videojuego Android para reforzar el aprendizaje de
vocabulario en idiomas

- Departamento: Ingeniería Informática y Filología Francesa e Inglesa
- Directores del proyecto: Manuel Palomo Duarte, Anke Berns
- Autor del proyecto: Adrián Pérez Fernández

Cádiz, 2 de diciembre de 2014

Fdo: Adrián Pérez Fernández

Agradecimientos

A mis padres y mi hermano, por todo el apoyo recibido. No solo en la realización de este proyecto, sino también a lo largo de la carrera.

A mis tutores, por sus incalculables consejos y aportaciones.

A los colaboradores, por su gran ayuda.

Licencia

Este documento ha sido liberado bajo Licencia GFDL 1.3 (GNU Free Documentation License). Se incluyen los términos de la licencia en inglés al final del mismo.

Copyright (c) 2014 Adrián Pérez Fernández.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Notación y formato

Cuando nos refiramos a un programa en concreto o librería, utilizaremos la notación:

LibGDX.

Cuando nos refiramos a un comando, o función de un lenguaje, usaremos la notación:

```
public void main(String [] args).
```


Índice general

1. Introducción	1
1.1. Motivación	1
1.2. Alcance	1
1.3. Sobre este documento	2
2. Descripción general del proyecto	3
2.1. Descripción del juego	3
2.2. Características de <i>MissingWords</i>	3
2.2.1. Concepto	3
2.2.2. Elementos del juego	4
2.3. Integrantes del proyecto	4
3. Planificación	5
3.1. Etapa inicial	5
3.2. Etapa de Aprendizaje	5
3.3. Etapa de desarrollo	7
3.4. Etapa de pruebas	7
3.5. Redacción de la memoria	7
3.6. Diagrama de Gantt	8
4. Análisis del sistema	9
4.1. Especificación de requisitos del sistema	9
4.1.1. Requisitos de interfaz externa	9
4.1.1.1. Requisitos de conexión de software y hardware	9
4.1.1.2. Interfaz de usuario	9
4.1.2. Requisitos funcionales	10
4.1.3. Requisitos de rendimiento	11
4.1.4. Requisitos del sistema software	11
4.2. Modelo de casos de uso	11
4.2.1. Diagrama de casos de uso	11
4.2.2. Descripción de los casos de uso	13
4.3. Modelo conceptual de datos	23
4.3.1. Diagrama Entidad-Relación	23
4.4. Modelo de comportamiento del sistema	26

4.4.1. Diagramas de secuencia y contrato de las operaciones del sistema	26
5. Diseño del sistema	59
5.1. Elementos gráficos	59
5.2. Elementos de sonido	59
5.3. Interfaz gráfica	59
5.4. Arquitectura general del sistema	72
5.4.1. Escenario	72
5.4.2. Actores	73
6. Implementación	75
6.1. Desarrollo	75
6.1.1. Desarrollo del sistema de palabras	75
6.1.2. Desarrollo de la lógica del juego	75
6.1.3. Desarrollo del minijuego	76
6.1.4. Unión de componentes	76
6.1.5. Mejoras e UI	76
6.2. Principales problemas de implementación	76
6.2.1. Soporte de múltiples tamaños de pantalla	76
6.2.2. Sistema multi-idioma	77
6.2.3. Conexión entre pantallas de juego	77
6.2.4. Comportamiento del jugador NPC	78
6.2.5. Datos persistentes	78
6.2.6. Codificación de ficheros	78
6.3. Sistema de control de versiones	79
6.4. Documentación del código	79
7. Pruebas	81
7.1. Pruebas unitarias	81
7.2. Pruebas de integración	82
7.3. Pruebas funcionales	82
7.4. Pruebas no funcionales	82
7.5. Pruebas de aceptación	82
8. Conclusiones	85
8.1. Experiencia personal	85
8.2. Mejoras futuras	86
Herramientas utilizadas	87
Manual de instalación	93
Manual de usuario	95
Manual del profesor	103
Bibliografía y referencias	107

GNU Free Documentation License	109
1. APPLICABILITY AND DEFINITIONS	109
2. VERBATIM COPYING	111
3. COPYING IN QUANTITY	111
4. MODIFICATIONS	111
5. COMBINING DOCUMENTS	113
6. COLLECTIONS OF DOCUMENTS	113
7. AGGREGATION WITH INDEPENDENT WORKS	114
8. TRANSLATION	114
9. TERMINATION	114
10. FUTURE REVISIONS OF THIS LICENSE	115
11. RELICENSING	115
ADDENDUM: How to use this License for your documents	115
 GNU General Public License	 117

Indice de figuras

3.1. Logo de LibGDX	6
3.2. Diagrama de Gantt	8
4.1. Diagrama de casos de uso	12
4.2. Diagrama Entidad-Relación	24
4.3. Diagrama de secuencia: Seleccionar Idioma	26
4.4. Diagrama de secuencia: Menú principal (escenario principal)	27
4.5. Diagrama de secuencia: Menú principal (escenario 2a)	28
4.6. Diagrama de secuencia: Menú principal (escenario 2b)	28
4.7. Diagrama de secuencia: Menú principal (escenario 2c)	29
4.8. Diagrama de secuencia: Menú principal (escenario 2d)	30
4.9. Diagrama de secuencia: Menú principal (escenario 2e)	30
4.10. Diagrama de secuencia: Jugador VS CPU	31
4.11. Diagrama de secuencia: Un jugador	32
4.12. Diagrama de secuencia: Seleccionar categoría (escenario principal)	32
4.13. Diagrama de secuencia: Seleccionar categoría (escenario 2a)	33
4.14. Diagrama de secuencia: Formación de palabras (escenario principal)	34
4.15. Diagrama de secuencia: Formación de palabras (escenario 4a)	37
4.16. Diagrama de secuencia: Formación de palabras (escenario 4b)	38
4.17. Diagrama de secuencia: Formación de palabras (escenario 5a)	39
4.18. Diagrama de secuencia: Formación de palabras (escenario 2-4)	40
4.19. Diagrama de secuencia: Minijuego (escenario principal y 4a)	41
4.20. Diagrama de secuencia: Minijuego (escenario 4b (2a))	43
4.21. Diagrama de secuencia: Minijuego (escenario 4b (2b))	44
4.22. Diagrama de secuencia: Minijuego (escenario 6c)	45
4.23. Diagrama de secuencia: Usar Pista (escenario principal)	46
4.24. Diagrama de secuencia: Usar Pista (escenario 1a)	47
4.25. Diagrama de secuencia: Usar Pista (escenario 1b)	48
4.26. Diagrama de secuencia: Pausar Juego (escenario principal)	48
4.27. Diagrama de secuencia: Pausar Juego (escenario 3a)	49
4.28. Diagrama de secuencia: Pausar Juego (escenario 3b)	50
4.29. Diagrama de secuencia: Reanudar	50
4.30. Diagrama de secuencia: Salir	51
4.31. Diagrama de secuencia: Estadísticas (escenario principal)	51

4.32. Diagrama de secuencia: Estadísticas (escenario 3a)	52
4.33. Diagrama de secuencia: Estadísticas Categoría X	53
4.34. Diagrama de secuencia: Ajustes (escenario principal)	54
4.35. Diagrama de secuencia: Ajustes (escenario 3a)	54
4.36. Diagrama de secuencia: Ajustes (escenario 3b)	55
4.37. Diagrama de secuencia: Activar/Desactivar Sonido	56
4.38. Diagrama de secuencia: Cambiar idioma	56
4.39. Diagrama de secuencia: Instrucciones	57
4.40. Diagrama de secuencia: Salir aplicación	58
5.1. Navegación de pantallas	60
5.2. Pantalla de selección de idioma	61
5.3. Pantalla de Menú principal	62
5.4. Pantalla de Selección de categoría	63
5.5. Pantalla de Estadísticas	64
5.6. Pantalla de Estadísticas por categoria	65
5.7. Pantalla de Ajustes	66
5.8. Pantalla de Instrucciones	67
5.9. Pantalla de juego: Formación de palabras	68
5.10. Pantalla de juego: Formación de palabras en pausa	69
5.11. Pantalla de juego: Minijuego	70
5.12. Pantalla de Victoria	71
5.13. Estructura de Scene2D	72

Capítulo 1

Introducción

1.1. Motivación

A lo largo de mi experiencia académica he tenido que hacer pequeños programas que me han sido muy útiles para aprender los conceptos básicos de la programación. Sin embargo, los resultados de dichas implementaciones se mostraban a través de la consola, lo cual estaba bien al principio, pero pasado un tiempo pensaba que debía superarme, es decir, darle una salida gráfica a todo el trabajo o con un estilo más “apropiado”.

Fue entonces cuando mi interés crecía cada vez más por la programación gráfica, tenía que haber algo más allá de la línea de comandos. Todo esto unido a mi afición por los juegos me llevaba a pensar que alguna vez podría hacer un videojuego.

Siempre he tenido claro que debía ser algo simple, como todo en la vida, hay que empezar desde abajo e ir subiendo. Por lo que pensé que ya era hora de saldar cuentas conmigo mismo y destronar a la rutinaria consola. Sin duda, ha sido una gran fuente de motivación para la creación de *MissingWords*.

1.2. Alcance

Como bien sabemos, los idiomas son una parte muy importante para nuestra formación académica. Hoy en día, donde la competencia es tan dura para conseguir entrar en el ámbito laboral, ser políglota es un punto muy fuerte a nuestro favor que puede determinar si el puesto es nuestro o no. Es por ello por lo que aprender nuevas lenguas es fundamental en nuestra preparación.

Cuando estudias un nuevo idioma, debes obtener las destrezas necesarias para demostrar que lo dominas. Para conseguir estas habilidades es evidente que hay que conocer el vocabulario propio y no es una tarea sencilla. Adquirir nuevas palabras es un proceso lento y que hay que ir afianzando poco a poco mediante el uso de las mismas.

MissingWords pretende ayudar a obtener este vocabulario que tanto esfuerzo requiere de forma amena y divertida, olvidándonos de que realmente estamos ante un juego educativo.

1.3. Sobre este documento

Este documento es la memoria del proyecto fin de carrera de la titulación *Ingeniería Técnica en Informática de Gestión* (ITIG) de la Universidad de Cádiz realizada por *Adrián Pérez Fernández*.

El documento está organizado en los siguientes capítulos:

1. *Introducción*: Resumen de la idea del proyecto así como la motivación que ha llevado a su realización.
2. *Descripción general del proyecto*: Descripción detallada del proyecto y que representa *MissingWords*, así como sus principales características.
3. *Planificación*: Organización temporal del desarrollo del proyecto.
4. *Análisis del sistema*: Análisis del sistema empleando la metodología *UML*. Comprende la definición de requisitos funcionales, diagramas de casos de uso y secuencia y contratos de operaciones.
5. *Diseño del sistema*: Descripción del diseño de la aplicación e interfaz de usuario.
6. *Implementación*: Detalles importantes de la implementación del proyecto. Principales problemas y como se han resuelto.
7. *Pruebas*: Pruebas realizadas para la verificación del software.
8. *Conclusiones*: Conclusiones sobre el proyecto, es decir, experiencia adquirida, reflexión personal y todo lo que ha conllevado su realización.
9. *Apéndices*: Herramientas utilizadas en la realización del proyecto, manuales de usuario, instalación y profesor, bibliografía y licencias.

Capítulo 2

Descripción general del proyecto

2.1. Descripción del juego

MissingWords es un videojuego de formación de palabras para aprender idiomas que nos permite adquirir nuevo vocabulario. Está disponible para Android y PC (Windows, Linux y Mac).

El juego consiste en formar palabras a partir de un conjunto de fichas (letras) que tendremos que ir seleccionando para crear una palabra perteneciente a la categoría que hayamos seleccionado y en un tiempo determinado. Opcionalmente, el juego ofrece 3 tipos de pistas: traducción, letra adicional y longitud. Obviamente, usar estas pistas conlleva una penalización. Una vez que consigues crear una palabra, el juego te otorga un número de tiradas (en base a tu construcción) que usarás en un minijuego. Por lo que tu objetivo es ganarlo ayudándote de la composición de vocablos.

Dispone de dos modos de juego: Un jugador y Jugador VS CPU. En el primero compites contra tí mismo y el tiempo. En el segundo modo, además del tiempo, tienes que vencer al sistema.

Por último, el jugador puede consultar el progreso que lleva mediante el menú estadísticas, tanto a nivel general (ratio de aciertos, partidas ganadas y perdidas, etc) como las palabras que le quedan para completar una categoría.

2.2. Características de *MissingWords*

El videojuego es totalmente libre y gratuito. Aunque el objetivo principal del juego es meramente educativo, también posee otro objetivo secundario que ayuda a entretener a la vez que aprendemos inconscientemente.

2.2.1. Concepto

Bien sabemos que un juego educativo como tal es una herramienta muy útil para aprender, pero el hecho de que sea totalmente didáctico hace que pierda el interés para la persona por no distraer, es decir, pierde el concepto de pasatiempo. Pasaría a ser otro método de enseñanza más,

que acabaría por ser abandonado.

Eso es lo que diferencia a *MissingWords* de otros juegos, el de abstraer al usuario de la formación de palabras (método de aprendizaje) para embarcarse a ganar el minijuego. Este concepto viene dado del “*Metagaming (juego dentro de otro juego)*”.

2.2.2. Elementos del juego

A continuación un breve resumen de los elementos más importantes del juego:

- **CPU:** Representa al jugador controlado por la IA.
- **Player:** Representa al usuario del juego.
- **Tile (Ficha):** Las distintas letras del abecedario representadas por fichas.
- **Estadísticas:** datos estadísticos de interés para el jugador. Se dividen en generales y por categorías.
- **Minijuego:** Juego paralelo a la constitución de palabras. Se trata de un tablero con un número de casillas determinado y agujeros generados al azar que los usuarios tendrán que sortear. Mediante un dado y las tiradas que tenemos disponibles haremos avanzar a nuestro personaje a través del camino.
- **Pistas:** ayudas opcionales para formar palabras más fácilmente. Conllevan penalizaciones.
- **Barra de tiempo:** Indica el tiempo que tenemos para formar una palabra.

2.3. Integrantes del proyecto

En este apartado se muestran las personas que han hecho posible este proyecto:

- **Diseño del juego:** Manuel Palomo Duarte, Anke Berns, Adrián Pérez Fernández.
- **Diseño de categorías e idiomas:**
 - **Alemán e Inglés:** Anke Berns, Alicia Garrido Guerrero, Andrea Calderón Márquez.
- **Programación:** Adrián Pérez Fernández.
- **Diseño gráfico:** Adrián Pérez Fernández.

Capítulo 3

Planificación

La planificación del desarrollo comprende varias etapas que se describen a continuación:

3.1. Etapa inicial

Aunque sabía que quería hacer un juego, no sabía de que tipo sería ni la complejidad que iba a suponer. Me puse en contacto con *Manuel Palomo Duarte* por recomendación de compañeros en referencia a los numerosos proyectos que había guiado en el tema de los videojuegos. Cuando hablamos me propuso la idea de hacer un juego educativo para aprender idiomas porque era en lo que estaba trabajando. Al principio pensé que un juego de tal calibre no sería todo lo interesante que yo imaginaba pero mas adelante descubriría que estaba equivocado. Contra todo pronóstico, acepté la propuesta con la motivación extra de que supondría un reto para mí. A partir de ahí, conocí a *Anke Berns*, la cual nos ayudaría a desarrollar la idea.

Pasé unos días reflexionando sobre ideas para el juego. Me apoyé en juegos que ya jugaba como *Apalabrados*, *Scrabble* o *Mezcladitos* como ejemplo a seguir. En la reunión que tuvimos describí mi idea de un juego de formación de palabras con algo más que un contexto educativo (tal como se detalló en la sección 2.2.1 del capítulo anterior), la cual nos pareció acertada. Así comenzó mi andanza en este proyecto fin de carrera.

3.2. Etapa de Aprendizaje

Esta etapa fue larga pero provechosa, aprendí muchas cosas mediante numerosos prototipos, que me dieron las bases para enfocar el proyecto.

- **Aprendizaje del SDK de Android y Java:** Al principio decidimos que sería para dispositivos móviles, por lo que estuve un tiempo aprendiendo a trabajar con el SDK oficial de Android. Hice el curso online *Programming Mobile Applications for Android Handheld Systems*¹ a través de la plataforma de educación virtual gratuita *Coursera* con el fin de

¹<https://www.coursera.org/course/android>

adquirir los conocimientos suficientes como para empezar a plantear el proyecto. Al mismo tiempo estuve aprendiendo Java, ya que el SDK de Android lo usaba como lenguaje de programación. Para mi sorpresa, no tuve muchos problemas en adaptarme viniendo de C++. Para su aprendizaje usé el entorno de desarrollo integrado *Eclipse*.

- **Elección del framework y aprendizaje de LibGDX:** Al mismo tiempo que hacía el curso e investigaba como abordar el proyecto, conocí *LibGDX*, un framework multiplataforma de código abierto para el desarrollo de videojuegos en 2D y 3D.



Figura 3.1: Logo de LibGDX

Me decanté finalmente por él debido a su gran adaptabilidad a otras plataformas, es decir, permite ejecutar el mismo código en distintos sistemas operativos sin apenas modificarlo. Asimismo, está muy bien documentado y posee una gran comunidad activa lanzando constantes actualizaciones y mejoras, lo que permite aprenderlo más fácilmente. Y como punto final, utiliza Java para su desarrollo, con el cual estaba empezando a familiarizarme. Por estas razones, *LibGDX* fue el mejor candidato para el desarrollo del videojuego.

- **Aprendizaje de la librería Scene2D:** Tan pronto como aprendí los fundamentos básicos de LibGDX, comencé a diseñar el juego. Me asaltaban muchas dudas sobre como debía estructurar el sistema. Estuve investigando como se solía hacer en este framework y pude observar que generalmente se seguían dos vertientes: los que hacían el juego con la librería Scene2D y los que no. Se supone que no todos los juegos son compatibles con las herramientas que recoge.

Scene2D gira en torno a dos elementos: el escenario y los actores. El escenario es la pantalla y los actores todos los objetos que se colocan en ella y que actúan en él. Me decanté por ella por su fácil entendimiento y simple estructura.

- **Aprendizaje de un software de control de versiones: Git:** A la vez que aprendía todo lo anterior me di cuenta que necesitaría un sistema de control de versiones para gestionar mi proyecto. El elegido fue Git. Consecuentemente la forja para alojar el código fuente sería GitHub, por su reputación y por estar basado en este.
- **Aprendizaje de herramientas gráficas:** Otro punto importante eran los gráficos. Me di cuenta que eran fundamentales para el juego y que no era nada fácil crear recursos de

calidad. Como nunca había trabajado con este tipo de programas, a través de tutoriales aprendí nociones básicas de Photoshop y Gimp.

3.3. Etapa de desarrollo

Esta etapa comprende todo el desarrollo del juego. Por consiguiente es la etapa más larga.

Al principio no tenía claro el tiempo que me llevaría terminar el software pero había estimado sobre unos seis u ocho meses. No obstante, no estaba totalmente centrado en el proyecto, puesto que tenía otras ocupaciones.

Parte de esta etapa y gran parte de la de aprendizaje estuve haciendo conjuntamente prácticas de empresa lo que hacía que todo fuera más lento durante los primeros meses del proyecto. El tener que adaptarse a un nuevo entorno y las responsabilidades que requerían las prácticas hacía que el tiempo que tenía para avanzar en el juego era menor. A todo esto hay que añadir que me estaba preparando el *First Certificate in English (University of Cambridge)*, lo cual hacía que tuviera que dividir aún más el tiempo. Al final todo se reducía a las tardes que tenía menos ocupaciones y fines de semana completos. Aunque fue lento me era de gran importancia tanto terminar las prácticas como obtener el certificado.

Cuando llevaba aproximadamente unos 3 meses, finalicé las prácticas y obtuve el certificado. A partir de ahí pude dedicarme plenamente al proyecto y empezar a trabajar muchas más horas diarias para alcanzar mi objetivo.

Fue por aquel entonces cuando retomaría la planificación establecida con el lapso de algunos meses por lo acontecido.

3.4. Etapa de pruebas

Probablemente la etapa más importante del proyecto. Con las pruebas comprobabas que realmente habías trasladado los conceptos aprendidos al proyecto. A medida que agregaba características nuevas, las iba probando al mismo tiempo para ver si todo funcionaba correctamente, y como no, aprender de mis errores.

Cuando creía que había hecho un cambio importante que debía de ser testeado más seriamente, lanzaba una versión nueva la cual era probada por *Anke Berns* y *Manuel Palomo* que me daban grandes consejos y críticas constructivas para mejorar la aplicación.

3.5. Redacción de la memoria

La realización de esta memoria ha tomado más importancia al final del desarrollo. Aunque anotaba los progresos y quedaban guardados en la forja, fue al final cuando dediqué mas tiempo a su confección.

3.6. Diagrama de Gantt

A continuación se muestra la planificación descrita anteriormente mediante el diagrama de Gantt:

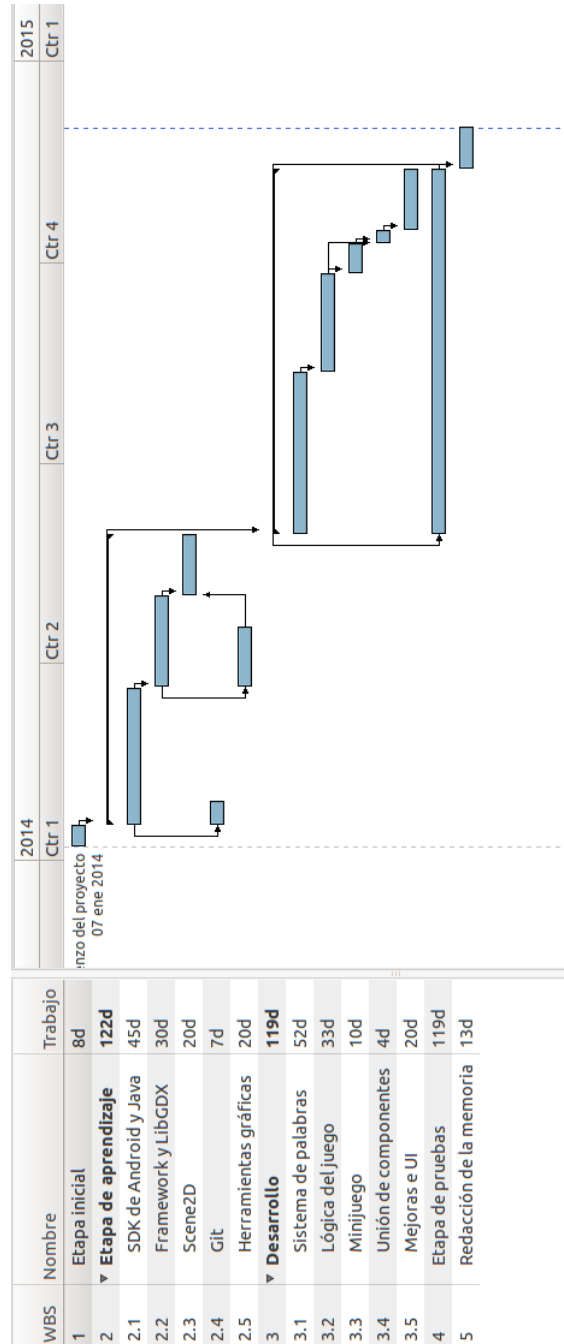


Figura 3.2: Diagrama de Gantt

Capítulo 4

Análisis del sistema

4.1. Especificación de requisitos del sistema

A continuación detallaremos los requisitos del sistema que el software ha de satisfacer así como las distintas condiciones que hay que establecer para su correcto funcionamiento.

4.1.1. Requisitos de interfaz externa

En este apartado se describirán los requisitos de conexión del software y el hardware, así como la interfaz de usuario

4.1.1.1. Requisitos de conexión de software y hardware

La conexión entre el software y el hardware dependerá de la plataforma donde se ejecute:

- Si se ejecuta bajo escritorio (Linux, Windows o Mac), de la conexión se encargará la JVM (*Java Virtual Machine*) de Java.
- Si se ejecuta en un dispositivo móvil (Android), de la conexión se encargará el SDK (*Software Development Kit*) de Android.

4.1.1.2. Interfaz de usuario

En cuanto a la interfaz del videojuego, su resolución óptima es 800x480 píxeles en escritorio. Aunque se adaptará a otras resoluciones, estrechando si es necesario pero sin afectar gravemente al diseño. En Android se ajustará a la resolución del dispositivo igualmente.

El usuario se comunicará con la aplicación mediante el uso exclusivo del ratón en escritorio y mediante la pantalla táctil en Android.

Las pantallas que componen el juego son las siguientes:

- **Selección de idioma:** Pantalla que permite al jugador seleccionar el idioma de juego.
- **Menú principal:** El menú principal que da paso a las siguientes pantallas de la aplicación:

- ◊ Jugador VS CPU
 - ◊ Un jugador
 - ◊ Estadísticas
 - ◊ Ajustes
 - ◊ Instrucciones
 - ◊ Salir
- **Selección de categoría:** Pantalla previa al juego que nos permite seleccionar la categoría a jugar. Se muestra cuando se escoge alguno de los dos modos de juego desde el menú principal: *Jugador VS CPU* o *Un jugador*.
 - **Estadísticas:** Muestra el progreso del jugador reflejando la información de todas sus partidas. Se dividen en estadísticas generales y estadísticas por categoría.
 - **Estadísticas categoría X:** Indica el progreso del jugador en una categoría en particular. Se accede desde el submenú *Estadísticas* del menú principal y luego haciendo clic en la categoría deseada.
 - **Ajustes:** Posibilita modificar los ajustes del juego: idioma y sonido. También se muestran los créditos del juego.
 - **Instrucciones:** Breve descripción del objetivo a conseguir y del significado de las pistas.
 - **Pantalla de juego:** Aquí se desarrollará el juego. Esta pantalla se dividirá en dos pantallas a su vez:
 - ✓ *Formación de palabras:* Aquí se llevará a cabo la composición de palabras. Tendremos disponible un tablero con 16 letras a usar para conseguir una palabra que esté dentro de la categoría seleccionada y en un tiempo determinado. Opcionalmente podremos usar pistas (con su penalización) mediante sus botones correspondientes que nos facilitaran la formación. En cuanto consigamos una, el sistema cambiará a la pantalla del minijuego.
 - ✓ *Minijuego:* En esta pantalla tendremos que hacer uso de las tiradas que conseguimos en la pantalla anterior. Mediante el uso de un dado moveremos nuestro personaje a través del tablero. Llegando a la meta, cumpliremos el objetivo y se nos proclamará ganador.
 - **Pantalla de victoria:** Pantalla de final de partida donde podemos ver un resumen de la partida. Esto comprende el número de turnos y palabras jugadas. También podemos ver las palabras que hemos formado de las totales que había en la categoría. Dentro de esta pantalla tenemos otras dos opciones: jugar de nuevo o volver al menú principal.

4.1.2. Requisitos funcionales

Los requisitos funcionales que debe cumplir el sistema son los siguientes:

- ✓ Seleccionar idioma de juego.

- ✓ Cambiar idioma de juego (una vez que estemos en un idioma concreto).
- ✓ Activar/desactivar el sonido.
- ✓ Consultar estadísticas.
- ✓ Ver las instrucciones.
- ✓ Seleccionar categoría de juego.
- ✓ Validar palabra.
- ✓ Salir de la aplicación en cualquier momento.

4.1.3. Requisitos de rendimiento

El sistema deberá dar una respuesta automática (los tiempos de respuesta no deben entorpecer enormemente el uso de la aplicación) a todas las acciones del usuario de forma que le ofrezca una experiencia satisfactoria.

Deben liberarse en todo momento los recursos de la memoria que no estén utilizándose para ahorrar batería (sobre todo cuando se ejecute bajo Android, donde los recursos móviles están más limitados).

4.1.4. Requisitos del sistema software

La aplicación cumplirá los siguientes requisitos del sistema software:

- ✓ Funcionará bajo los entornos de escritorio Windows, Linux y Mac que cuenten con la versión de Java 7 instalada.
- ✓ Funcionará como mínimo en la versión 2.3.x de Android dando soporte a más del 92 % de dispositivos según Google en el mes de Noviembre de 2014.

4.2. Modelo de casos de uso

Para representar los requisitos del sistema usaremos el estándar *UML* (Unified Modeling Language), así como para la representación de los diagramas.

4.2.1. Diagrama de casos de uso

A continuación se muestra el diagrama de casos de uso de la aplicación:

4.2.2. Descripción de los casos de uso

Seguidamente pasamos a la descripción de los casos de uso del diagrama anterior (figura 4.1). Detallaremos los escenarios usando la plantilla según el modelo *UML*.

Caso de uso: Seleccionar idioma

Nombre: Seleccionar idioma

Descripción: El usuario escoge el idioma de juego inicial

Actores: Usuario

Precondiciones: Ninguna

Postcondiciones: La aplicación se inicia para el idioma seleccionado

Escenario principal:

1. El sistema muestra en la pantalla banderas con los idiomas disponibles.
2. El usuario selecciona la bandera del idioma deseado.
3. El sistema guarda el idioma seleccionado y muestra el menú principal.

Caso de uso: Menú principal

Nombre: Menú principal

Descripción: Se muestra el menú principal de la aplicación.

Actores: Usuario

Precondiciones: Ninguna

Postcondiciones: Ninguna

Escenario principal:

1. El sistema muestra el menú principal.
2. El usuario selecciona la opción *Jugador VS CPU*.
3. «include» **Jugador VS CPU**.

Extensiones (flujos alternativos):

- 2a. El usuario selecciona la opción *Un jugador*.
 1. «include» **Un jugador**.
- 2b. El usuario selecciona la opción *Estadísticas*.
 1. «include» **Estadísticas**.

- 2c. El usuario selecciona la opción *Ajustes*.
 - 1. «include» **Ajustes**.
- 2d. El usuario selecciona la opción *Instrucciones*.
 - 1. «include» **Instrucciones**.
- 2e. El usuario selecciona la opción *Salir*.
 - 1. «include» **Salir aplicación**.

Caso de uso: Jugador VS CPU

Nombre: Jugador VS CPU

Descripción: Se selecciona el modo *Jugador VS CPU* para empezar un nuevo juego

Actores: Usuario

Precondiciones: El usuario se encuentra en el menú principal

Postcondiciones: Ninguna

Escenario principal:

- 1. El usuario pulsa sobre el botón *Jugador VS CPU* del menú principal.
- 2. «include» **Seleccionar categoría**.

Caso de uso: Un jugador

Nombre: Un jugador

Descripción: Se selecciona el modo *Un jugador* para empezar un nuevo juego

Actores: Usuario

Precondiciones: El usuario se encuentra en el menú principal

Postcondiciones: Ninguna

Escenario principal:

- 1. El usuario pulsa sobre el botón *Un jugador* del menú principal.
- 2. El sistema activa el modo un jugador.
- 3. «include» **Seleccionar categoría**.

Caso de uso: Seleccionar categoría

Nombre: Seleccionar categoría

Descripción: Se selecciona una categoría para empezar el juego

Actores: Usuario

Precondiciones: Se ha seleccionado alguno de los modos de juego disponibles

Postcondiciones: Se empieza un nuevo juego con el modo de juego seleccionado

Escenario principal:

1. El sistema muestra las categorías disponibles para jugar.
2. El usuario selecciona una de las categorías.
3. El sistema guarda la categoría seleccionada e inicia el juego. «*include*» **Formación de palabras.**

Extensiones (flujos alternativos):

- 2a. El usuario se arrepiente y decide volver al menú principal.
 1. El usuario pulsa el botón *Volver atrás*.
 2. «*include*» **Menú principal.**

Caso de uso: Formación de palabras

Nombre: Formación de palabras

Descripción: Una de las pantallas de juego (junto con la del minijuego) donde el objetivo es conseguir formar una palabra

Actores: Usuario

Precondiciones: Ninguna

Postcondiciones: El usuario acaba una partida

Escenario principal:

1. El sistema carga los recursos necesarios. Muestra la información de las tiradas, los botones de pistas y validación y el tiempo restante.
2. El sistema comienza un nuevo turno y muestra un mensaje que indica de quién es el turno actual (Si estamos en el modo *Un jugador*, siempre jugará el usuario)
3. El sistema genera un cuadro con 16 fichas y las muestra.
4. El sistema activa el tiempo.
5. El usuario (o CPU si jugamos en el modo *Jugador VS CPU* y es su turno) selecciona fichas hasta conseguir una palabra y pulsa en el botón de validación de palabra. Al mismo tiempo el sistema suma las letras que se están escogiendo para mostrar la puntuación de la palabra que está formando.

6. El sistema comprueba que la palabra sea correcta y asigna las tiradas en base a los puntos de la palabra creada e inicia el minijuego. «*include*» **Minijuego**.
 7. El sistema que controla el caso de uso de la formación de palabras queda a la espera de que se utilicen las tiradas en el minijuego.
 8. El minijuego termina indicando que no se ha logrado el objetivo y el caso de uso **Formación de palabras** toma el control nuevamente. El sistema muestra el feedback positivo por haber terminado el minijuego.
 9. El sistema finaliza el turno y da comienzo a uno nuevo (cediendo el turno en el caso del modo *Jugador VS CPU*).
- * Repetir los **pasos 2-9** hasta que se logre el objetivo del minijuego.

Extensiones (flujos alternativos):

- 4a. El usuario decide usar una pista.
 1. «*extend*» **Usar pista**.
- 4b. El tiempo se agota.
 1. El sistema reproduce un sonido indicando que el tiempo se ha agotado y finaliza el turno (**Paso 8**).
- 5a. La palabra introducida no pertenece a la categoría que se está jugando.
 1. El sistema muestra un mensaje al usuario de que no se ha encontrado la palabra y continúa el juego.
- 2-4 El usuario decide pausar el juego.
 1. «*extend*» **Pausar juego**.

Caso de uso: Minijuego

Nombre: Minijuego

Descripción: Minijuego para gastar las tiradas que hemos conseguido en el caso de uso formación de palabras

Actores: Usuario

Precondiciones: Existe una partida en curso y se ha formado una palabra correctamente

Postcondiciones: El usuario acaba una partida

Escenario principal:

1. El sistema muestra el tablero del minijuego.
2. El usuario (o CPU si estamos en el modo *Player VS CPU*) pulsa el botón *Tirar* para lanzar un dado y ver el resultado.

3. El sistema decrementa el número de tiradas.
 4. El usuario pulsa el botón *Mover* para que su personaje avance.
 5. El sistema mueve el personaje tantas casillas como indica la tirada.
 6. El usuario (o CPU) alcanza la meta.
 7. El sistema finaliza la partida y muestra la pantalla de victoria.
- * Repetir los **pasos 2-5** hasta que se agoten las tiradas o se llegue a la meta (**paso 6**).

Extensiones (flujos alternativos):

- 4a. El usuario decide no mover porque no le agrada el resultado y todavía tiene tiradas restantes.
 1. El jugador vuelve a tirar el dado (**paso 2**).
- 4b. El usuario decide no mover porque no le agrada el resultado y no le quedan tiradas.
 1. El jugador pulsa el botón *Esperar*.
 2. El sistema comprueba que el usuario ha movido al menos una de sus tiradas.
 - 2a. El usuario ha movido al menos una de sus tiradas.
 1. El sistema muestra el botón *Continuar*.
 2. El usuario pulsa el botón.
 3. El sistema vuelve a ceder el control al caso de uso **Formación de palabras** desde el punto en que lo dejó.
 - 2b. El usuario no ha movido ninguna de sus tiradas.
 1. El sistema indica al usuario mediante un mensaje que debe mover al menos una de sus tiradas.
 2. El usuario debe pulsar el botón *Mover* y lo hace (**paso 4**).
- 6a. El usuario (o CPU) no alcanza la meta y tiene tiradas restantes.
 1. El usuario (o CPU) vuelve a tirar de nuevo (**paso 2**).
- 6b. El usuario (o CPU) no alcanza la meta y no tiene tiradas restantes.
 1. El sistema muestra el botón *Continuar*
 2. El usuario pulsa el botón.
 3. El sistema vuelve a ceder el control al caso de uso **Formación de palabras** desde el punto en que lo dejó.
- 6c. El usuario (o CPU) cae en un agujero.
 1. El sistema coloca el personaje en la casilla de salida correspondiente y la partida continúa.
 2. Se procede con el **paso 6a** o **6b** según si se tienen tiradas.

Caso de uso: Usar pista

Nombre: Usar pista

Descripción: Se usa una pista como apoyo a la composición de palabras.

Actores: Usuario

Precondiciones: Existe una partida en curso

Postcondiciones: El sistema da una pista

Escenario principal:

1. El usuario pulsa el botón de pista traducción y el sistema traduce al castellano una de las palabras que se encuentre en el tablero y muestra dicha traducción.
2. El sistema aplica las penalizaciones correspondientes y las muestra debajo del contador de puntuación.

Extensiones (flujos alternativos):

- 1a. El usuario pulsa el botón de pista letra adicional.
 1. El sistema selecciona la primera letra de una de las palabras del tablero.
- 1b. El usuario pulsa el botón de pista longitud.
 1. El sistema muestra la longitud de una de las palabras del tablero mediante fichas con interrogaciones.

Caso de uso: Pausar juego

Nombre: Pausar juego

Descripción: Pausa el juego a elección del usuario

Actores: Usuario

Precondiciones: Existe una partida en curso

Postcondiciones: La partida se pausa

Escenario principal:

1. El usuario pulsa el botón de pausa.
2. El sistema pausa el tiempo, oculta el tablero de letras y muestra las opciones del menú de pausa.
3. El sistema queda en espera hasta nuevo aviso.

Extensiones (flujos alternativos):

- 3a. El usuario decide reanudar la partida.
 1. «include» **Reanudar.**
- 3b. El usuario decide salir de la partida.

1. «include» **Salir partida.**

Caso de uso: Reanudar

Nombre: Pausar juego

Descripción: Reanuda la partida para continuar jugando

Actores: Usuario

Precondiciones: Existe una partida en curso y está pausada

Postcondiciones: La partida se reanuda

Escenario principal:

1. El usuario pulsa el botón de *Reanudar*.
2. El sistema activa el tiempo y muestra el tablero de letras.

Caso de uso: Salir partida

Nombre: Salir partida

Descripción: Abandona la partida que se está jugando

Actores: Usuario

Precondiciones: Existe una partida en curso y está pausada

Postcondiciones: La partida se acaba perdiendo los progresos y se vuelve al menú principal

Escenario principal:

1. El usuario pulsa el botón de *Salir*.
2. El sistema cancela la partida y muestra el menú principal. «include» **Menú principal.**

Caso de uso: Estadísticas

Nombre: Estadísticas

Descripción: Muestra la pantalla de estadísticas

Actores: Usuario

Precondiciones: El usuario se encuentra en el menú principal

Postcondiciones: Ninguna

Escenario principal:

1. El usuario pulsa el botón *Estadísticas* del menú principal.
2. El sistema muestra la pantalla de estadísticas.
3. El usuario consulta las estadísticas generales si lo desea.

Extensiones(flujos alternativos):

- 3a. El usuario desea consultar las estadísticas de una categoría determinada.
 1. «include» **Mostrar estadísticas categoría X.**
- 3b. El usuario termina de consultar las estadísticas.
 1. El usuario pulsa el botón *Volver atrás*.
 2. «include» **Menú principal.**

Caso de uso: Mostrar estadísticas categoría X

Nombre: Mostrar estadísticas categoría X

Descripción: Muestra la pantalla de estadísticas de la categoría seleccionada

Actores: Usuario

Precondiciones: El usuario se encuentra en la pantalla de estadísticas

Postcondiciones: Ninguna

Escenario principal:

1. El usuario pulsa en una categoría para ver sus estadísticas.
2. El sistema muestra las estadísticas de la categoría.
3. El usuario consulta las estadísticas de la categoría.

Extensiones(flujos alternativos):

- 3a. El usuario termina de consultar las estadísticas de la categoría
 1. El usuario pulsa el botón *Volver atrás*.
 2. «include» **Estadísticas.**

Caso de uso: Ajustes

Nombre: Ajustes

Descripción: Muestra la pantalla de ajustes

Actores: Usuario

Precondiciones: El usuario se encuentra en el menú principal

Postcondiciones: Ninguna

Escenario principal:

1. El usuario pulsa en el botón *Ajustes* del menú principal.
2. El sistema muestra los ajustes de la aplicación.
3. El usuario consulta los ajustes.

Extensiones(flujos alternativos):

- 3a. El usuario desea activar/desactivar el sonido de la aplicación.
 1. «*include*» **Activar/Desactivar sonido.**
- 3b. El usuario desea cambiar el idioma del juego.
 1. «*include*» **Cambiar idioma.**
- 3c. El usuario termina de consultar los ajustes.
 1. El usuario pulsa el botón *Volver atrás*.
 2. «*include*» **Menú principal.**

Caso de uso: Activar/desactivar sonido

Nombre: Activar/desactivar sonido

Descripción: Activa o desactiva el sonido de la aplicación

Actores: Usuario

Precondiciones: El usuario se encuentra en la pantalla de ajustes

Postcondiciones: Ninguna

Escenario principal:

1. El usuario marca o desmarca el checkbox dependiendo de si quiere activar o desactivar el sonido.
2. El sistema guarda los ajustes de sonido y los aplica.

Caso de uso: Cambiar idioma

Nombre: Cambiar idioma

Descripción: Cambia el idioma de juego

Actores: Usuario

Precondiciones: El usuario se encuentra en la pantalla de ajustes

Postcondiciones: Ninguna

Escenario principal:

1. El usuario selecciona la bandera del idioma al que desee cambiar.
2. El sistema guarda los ajustes de idioma y los aplica.

Caso de uso: Instrucciones

Nombre: Instrucciones

Descripción: Muestra la pantalla de instrucciones

Actores: Usuario

Precondiciones: El usuario se encuentra en el menú principal

Postcondiciones: Ninguna

Escenario principal:

1. El usuario pulsa en el botón *Instrucciones* del menú principal.
2. El sistema muestra las instrucciones del juego.
3. El usuario consulta las instrucciones.

Extensiones(flujos alternativos):

- 3a. El usuario termina de consultar las instrucciones.
 1. El usuario pulsa el botón *Volver atrás*.
 2. «include» **Menú principal**.

Caso de uso: Salir aplicación

Nombre: Salir aplicación

Descripción: Sale de la aplicación

Actores: Usuario

Precondiciones: El usuario se encuentra en el menú principal

Postcondiciones: Sale de la aplicación

Escenario principal:

1. El usuario pulsa en el botón *Salir* del menú principal.
2. El sistema libera los recursos y finaliza la aplicación.

4.3. Modelo conceptual de datos

En el modelo conceptual de datos nos centramos en describir las estructuras de datos, clases que intervienen y las relaciones que hay entre ellas.

Para ello utilizaremos el diagrama Entidad-Relación y la notación *UML*, tal como hemos hecho en el Modelo de casos de uso.

4.3.1. Diagrama Entidad-Relación

A continuación mostramos el diagrama ER para las clases más importantes utilizadas en la implementación, así como sus relaciones.

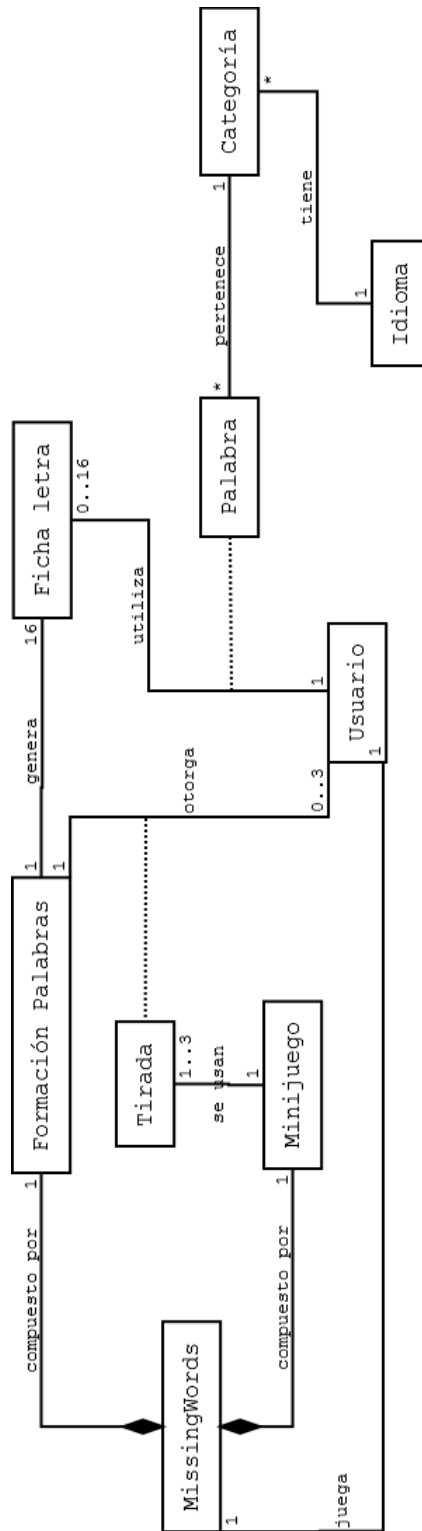


Figura 4.2: Diagrama Entidad-Relación

Restricciones Textuales:

- *Tirada* y *Palabra* son atributos de enlace.
- El atributo de enlace *Tirada* se otorga en base a los puntos que se obtienen de una palabra.

Entidades y relaciones:

■ **Entidades:**

- *MissingWords*: Entidad que representa el juego en cuestión.
- *Formación Palabras*: Entidad que compone el juego. Constituye el sistema de formación de palabras.
- *Minijuego*: Entidad que compone el juego. Minijuego que hay que superar para ganar.
- *Usuario*: Usuario que juega.
- *Ficha Letra*: Cada una de las fichas que conforman el tablero para formar palabras.
- *Categoría*: Entidad que representa las categorías del juego.
- *Idioma*: Idiomas disponibles del juego.
- *Tirada*: Atributo de enlace. Se le otorgan al jugador cuando realiza una palabra. Se usan en el minijuego.
- *Palabra*: Atributo de enlace. El jugador debe formar palabras haciendo uso de las fichas de letras.

■ **Relaciones:**

- *Compuesto por*: Relación de composición. Indica que el juego no tendría sentido sin sus componentes. *MissingWords* está compuesto de la formación de palabras y el minijuego.
- *Juega*: Relación 1 : 1. Un solo usuario puede jugar a una única instancia del juego.
- *Genera*: Relación 1 : 16. La formación de palabras ofrece 16 letras con las que jugar.
- *Otorga*: Relación 1 : 0..3. Por una palabra se otorgan de 0 a 3. 0 si no hace palabra. De 1 a 3 en base a la construcción.
- *Se usan*: Relación 1 : 3. Tiradas que se usan en el minijuego.
- *Utiliza*: Relación 1 : 0..16. El usuario utiliza n fichas para formar
- *Pertenece*: Relación 1 : N. Una categoría tiene N palabras.
- *Tiene*: Relación 1 : N. Un idioma tiene N categorías.

4.4. Modelo de comportamiento del sistema

El modelo de comportamiento describe como actúa el sistema. Consta de dos partes:

- Diagramas de secuencia del sistema: presenta la secuencia de eventos entre los actores y el sistema.
- Contrato de las operaciones del sistema: Describen el efecto de las operaciones del sistema.

4.4.1. Diagramas de secuencia y contrato de las operaciones del sistema

Mostraremos seguidamente los diagramas de secuencia y operaciones más importantes del sistema mediante la notación *UML*. Por claridad, suprimiremos las operaciones que ya hayan aparecido en algún punto.

Caso de uso: Seleccionar idioma

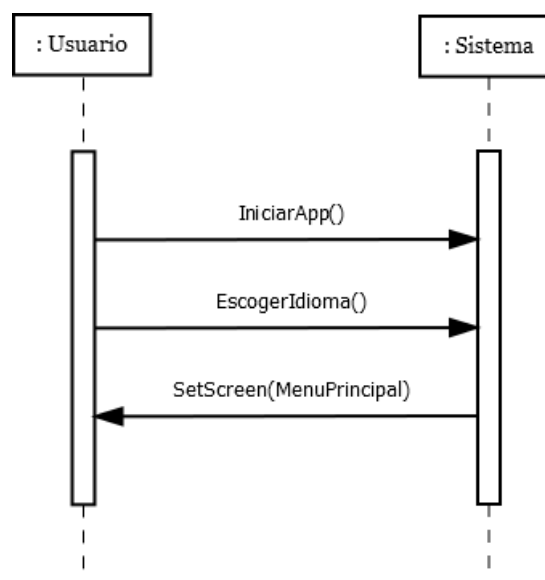


Figura 4.3: Diagrama de secuencia: Seleccionar Idioma

Operación: IniciarApp()

- *Actores:* Usuario, Sistema.
- *Responsabilidades:* Iniciar la aplicación y mostrar la pantalla de selección de idioma.
- *Referencias cruzadas:* Caso de uso: Seleccionar idioma.
- *Precondiciones:* Ninguna.
- *Postcondiciones:* Se inicia la aplicación.

Operación: EscogerIdioma()

- *Actores:* Usuario, Sistema.
- *Responsabilidades:* Establecer el idioma de juego.
- *Referencias cruzadas:* Caso de uso: Seleccionar idioma.
- *Precondiciones:* Se ha iniciado la aplicación.
- *Postcondiciones:* Guarda el idioma de juego y lo establece.

Operación: SetScreen(screen)

- *Actores:* Sistema.
- *Responsabilidades:* Muestra la pantalla indicada.
- *Referencias cruzadas:* Caso de uso: Seleccionar idioma.
- *Precondiciones:* Ninguna.
- *Postcondiciones:* Se muestra la pantalla recibida como parámetro.

Caso de uso: Menú principal (escenario principal)

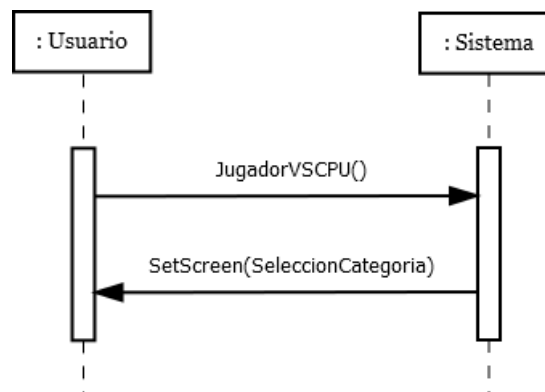


Figura 4.4: Diagrama de secuencia: Menú principal (escenario principal)

Operación: JugadorVSCPU()

- *Actores:* Usuario, Sistema.
- *Responsabilidades:* Cargar la pantalla de selección de categoría.
- *Referencias cruzadas:* Caso de uso: Menú principal
- *Precondiciones:* El usuario se encontraba en el menú principal y ha seleccionado la opción Jugador VS CPU.
- *Postcondiciones:* El sistema carga la pantalla de selección de categoría.

Caso de uso: Menú principal (escenario 2a)

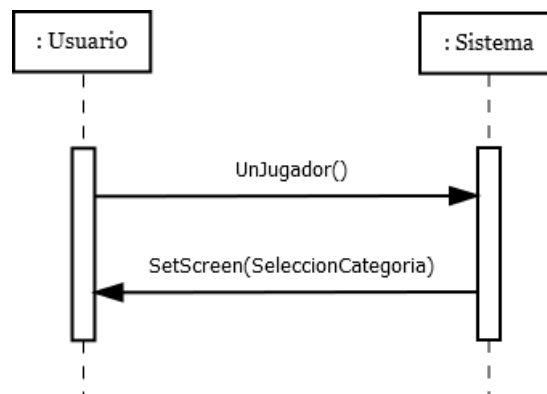


Figura 4.5: Diagrama de secuencia: Menú principal (escenario 2a)

Operación: `UnJugador()`

- *Actores:* Usuario, Sistema.
- *Responsabilidades:* Activar el modo de un jugador y cargar la pantalla de selección de categoría.
- *Referencias cruzadas:* Caso de uso: Menú principal
- *Precondiciones:* El usuario se encontraba en el menú principal y ha seleccionado la opción Un Jugador.
- *Postcondiciones:* El sistema activa el modo un jugador y carga la pantalla de selección de categoría.

Caso de uso: Menú principal (escenario 2b)

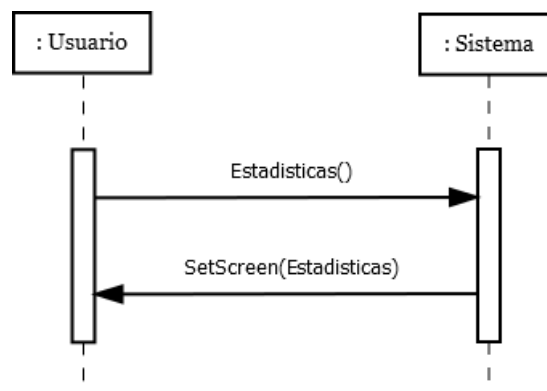


Figura 4.6: Diagrama de secuencia: Menú principal (escenario 2b)

Operación: Estadísticas()

- *Actores:* Usuario, Sistema.
- *Responsabilidades:* Cargar la pantalla de estadísticas.
- *Referencias cruzadas:* Caso de uso: Menú principal
- *Precondiciones:* El usuario se encontraba en el menú principal y ha seleccionado la opción Estadísticas.
- *Postcondiciones:* El sistema carga la pantalla de estadísticas.

Caso de uso: Menú principal (escenario 2c)

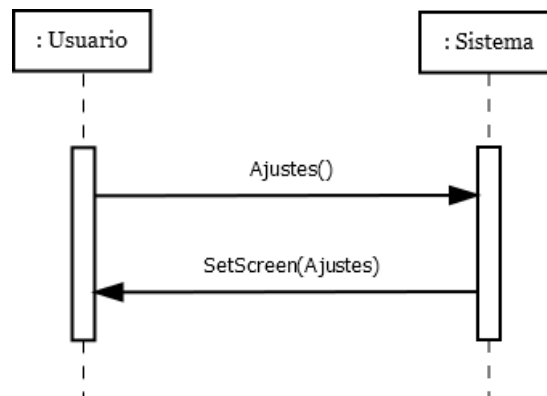


Figura 4.7: Diagrama de secuencia: Menú principal (escenario 2c)

Operación: Ajustes()

- *Actores:* Usuario, Sistema.
- *Responsabilidades:* Cargar la pantalla de ajustes.
- *Referencias cruzadas:* Caso de uso: Menú principal
- *Precondiciones:* El usuario se encontraba en el menú principal y ha seleccionado la opción Ajustes.
- *Postcondiciones:* El sistema carga la pantalla de ajustes.

Caso de uso: Menú principal (escenario 2d)

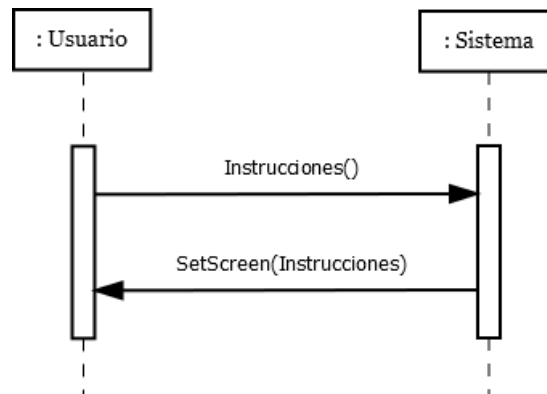


Figura 4.8: Diagrama de secuencia: Menú principal (escenario 2d)

Operación: Instrucciones()

- *Actores:* Usuario, Sistema.
- *Responsabilidades:* Cargar la pantalla de instrucciones.
- *Referencias cruzadas:* Caso de uso: Menú principal
- *Precondiciones:* El usuario se encontraba en el menú principal y ha seleccionado la opción Instrucciones.
- *Postcondiciones:* El sistema carga la pantalla de instrucciones.

Caso de uso: Menú principal (escenario 2e)

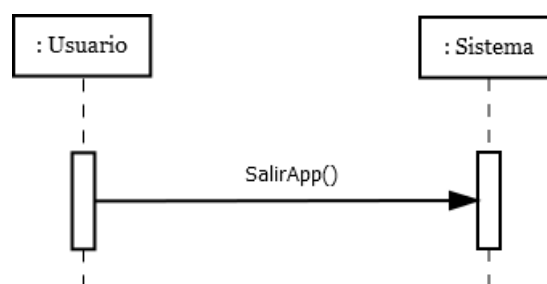


Figura 4.9: Diagrama de secuencia: Menú principal (escenario 2e)

Operación: SalirApp()

- *Actores:* Usuario, Sistema.
- *Responsabilidades:* Indica al sistema terminar con la aplicación.

- *Referencias cruzadas:* Caso de uso: Menú principal
- *Precondiciones:* El usuario se encontraba en el menú principal y ha seleccionado la opción Salir.
- *Postcondiciones:* El sistema indica que se desea salir de la aplicación.

Caso de uso: Player VS CPU

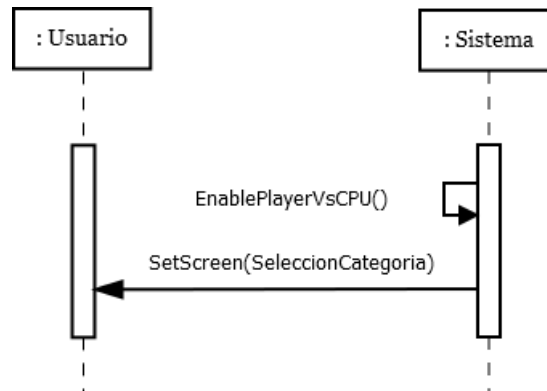


Figura 4.10: Diagrama de secuencia: Jugador VS CPU

Operación: EnablePlayerVSCPU()

- *Actores:* Sistema.
- *Responsabilidades:* Activar el modo de Jugador VS CPU.
- *Referencias cruzadas:* Caso de uso: Jugador VS CPU
- *Precondiciones:* El usuario se encontraba en el menú principal y ha seleccionado la opción Jugador VS CPU.
- *Postcondiciones:* El sistema activa el modo de Jugador VS CPU.

Caso de uso: Un Jugador

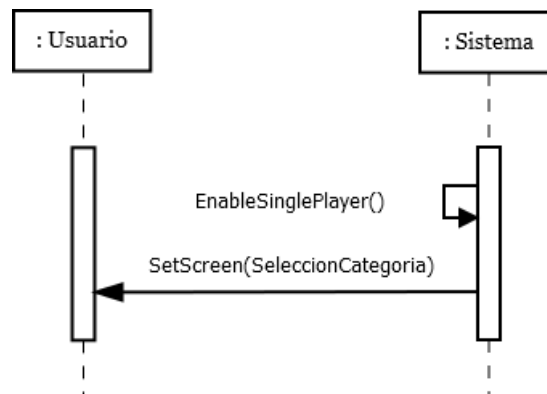


Figura 4.11: Diagrama de secuencia: Un jugador

Operación: EnableSinglePlayer()

- *Actores:* Sistema.
- *Responsabilidades:* Activar el modo de Un jugador.
- *Referencias cruzadas:* Caso de uso: Un jugador.
- *Precondiciones:* El usuario se encontraba en el menú principal y ha seleccionado la opción Un jugador.
- *Postcondiciones:* El sistema activa el modo de Un jugador.

Caso de uso: Seleccionar categoría (escenario principal)

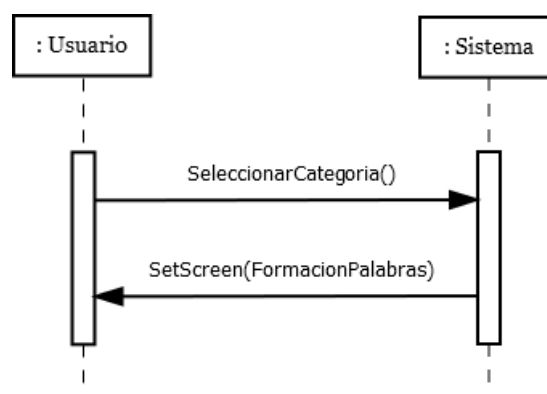


Figura 4.12: Diagrama de secuencia: Seleccionar categoría (escenario principal)

Operación: SeleccionarCategoria()

- *Actores:* Usuario, Sistema.
- *Responsabilidades:* Guardar la categoría seleccionada por el usuario.
- *Referencias cruzadas:* Caso de uso: Seleccionar categoría.
- *Precondiciones:* El usuario selecciona una categoría.
- *Postcondiciones:* El sistema guarda la categoría para la partida.

Caso de uso: Seleccionar categoría (escenario 2a)

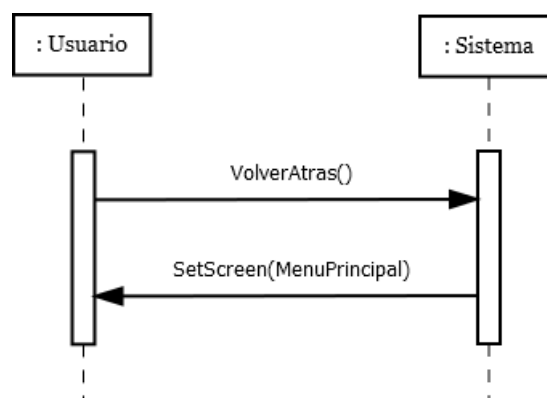


Figura 4.13: Diagrama de secuencia: Seleccionar categoría (escenario 2a)

Operación: VolverAtras()

- *Actores:* Usuario, Sistema.
- *Responsabilidades:* Devolver al usuario a la pantalla anterior.
- *Referencias cruzadas:* Caso de uso: Seleccionar categoría.
- *Precondiciones:* El usuario pulsa el botón volver atras.
- *Postcondiciones:* El sistema le devuelve a la pantalla anterior.

Caso de uso: Formación de palabras (escenario principal)

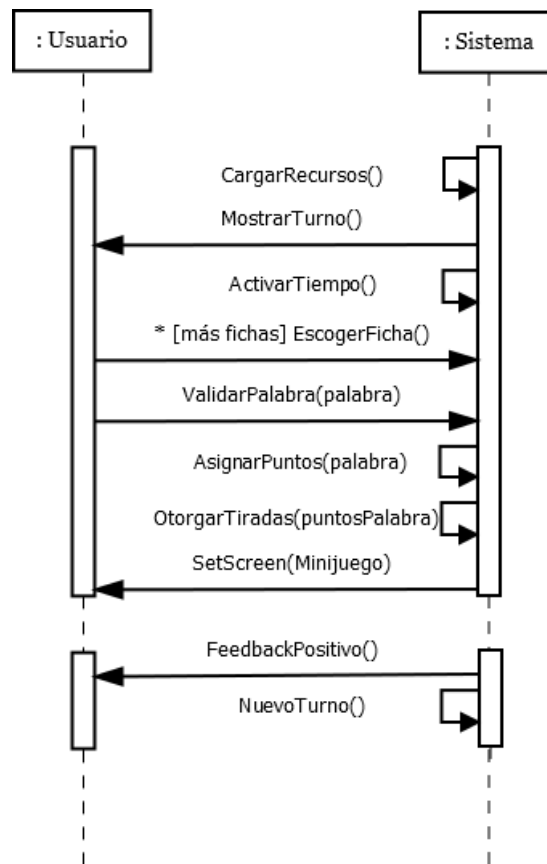


Figura 4.14: Diagrama de secuencia: Formación de palabras (escenario principal)

Operación: CargarRecursos()

- *Actores:* Sistema.
- *Responsabilidades:* Cargar los recursos necesarios de la pantalla.
- *Referencias cruzadas:* Caso de uso: Formación de palabras.
- *Precondiciones:* El usuario ha empezado un juego.
- *Postcondiciones:* El carga los recursos y los muestra.

Operación: MostrarTurno()

- *Actores:* Sistema.
- *Responsabilidades:* Muestra al usuario quién juega en el turno actual.
- *Referencias cruzadas:* Caso de uso: Formación de palabras.

- *Precondiciones*: El usuario ha empezado un juego.
- *Postcondiciones*: Muestra al jugador quién juega en este turno.

Operación: ActivarTiempo()

- *Actores*: Sistema.
- *Responsabilidades*: Activar el tiempo para la formación de palabras.
- *Referencias cruzadas*: Caso de uso: Formación de palabras.
- *Precondiciones*: El usuario ha empezado un juego.
- *Postcondiciones*: Activa el tiempo para la formación de palabras.

Operación: EscogerFicha()

- *Actores*: Usuario, Sistema.
- *Responsabilidades*: El usuario selecciona una ficha.
- *Referencias cruzadas*: Caso de uso: Formación de palabras.
- *Precondiciones*: El usuario ha empezado un juego.
- *Postcondiciones*: El sistema aparta la ficha del bloque para la formación.

Operación: ValidarPalabra(palabra)

- *Actores*: Usuario, Sistema.
- *Responsabilidades*: Validar la palabra enviada.
- *Referencias cruzadas*: Caso de uso: Formación de palabras.
- *Precondiciones*: El usuario ha empezado un juego y ha enviado una palabra.
- *Postcondiciones*: Comprueba que la palabra es correcta y pertenece a la categoría seleccionada.

Operación: AsignarPuntos(palabra)

- *Actores*: Sistema.
- *Responsabilidades*: Asignar puntos en función de la palabra formada.
- *Referencias cruzadas*: Caso de uso: Formación de palabras.
- *Precondiciones*: El usuario ha empezado un juego y ha formado una palabra.
- *Postcondiciones*: Asigna los puntos conseguidos por la palabra.

Operación: OtorgarTiradas(puntosPalabra)

- *Actores:* Sistema.
- *Responsabilidades:* Otorgar tiradas en función de los puntos conseguidos.
- *Referencias cruzadas:* Caso de uso: Formación de palabras.
- *Precondiciones:* El usuario ha empezado un juego y ha formado una palabra.
- *Postcondiciones:* Otorga las tiradas pertinentes al jugador.

Operación: FeedbakPositivo()

- *Actores:* Sistema.
- *Responsabilidades:* Mostrar al usuario un feedback positivo por completar el minijuego.
- *Referencias cruzadas:* Caso de uso: Formación de palabras.
- *Precondiciones:* El usuario ha empezado un juego y ha gastado sus tiradas en el minijuego.
- *Postcondiciones:* Muestra al jugador un feedback positivo.

Operación: NuevoTurno()

- *Actores:* Sistema.
- *Responsabilidades:* Iniciar un nuevo turno.
- *Referencias cruzadas:* Caso de uso: Formación de palabras.
- *Precondiciones:* El usuario ha empezado un juego y ha terminado su turno.
- *Postcondiciones:* El sistema inicia un nuevo turno.

Caso de uso: Formación de palabras (escenario 4a)

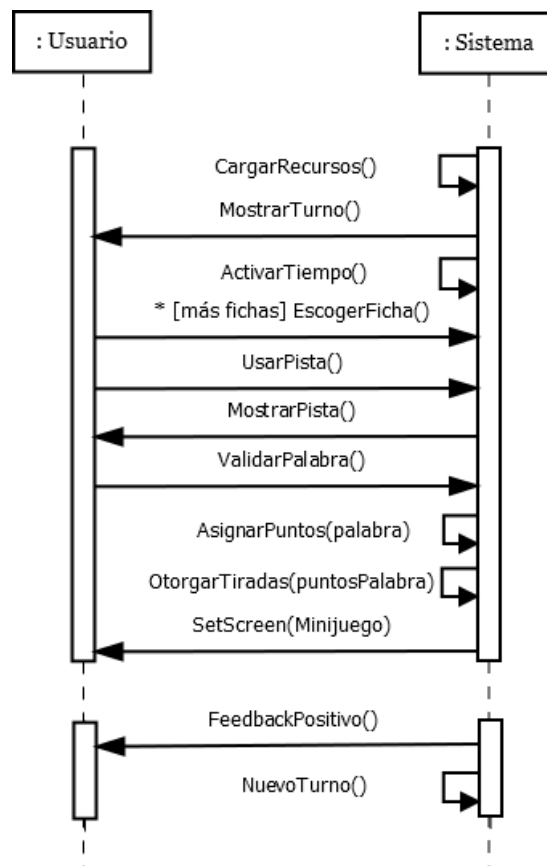


Figura 4.15: Diagrama de secuencia: Formación de palabras (escenario 4a)

Operación: UsarPista()

- *Actores:* Usuario.
- *Responsabilidades:* El usuario decide usar una pista.
- *Referencias cruzadas:* Caso de uso: Formación de palabras.
- *Precondiciones:* El usuario ha empezado un juego.
- *Postcondiciones:* Ninguna.

Operación: MostrarPista()

- *Actores:* Sistema.
- *Responsabilidades:* Mostrar pista al usuario.
- *Referencias cruzadas:* Caso de uso: Formación de palabras.
- *Precondiciones:* El usuario ha empezado un juego y ha pedido una pista.
- *Postcondiciones:* El sistema muestra la pista correspondiente al usuario.

Caso de uso: Formación de palabras (escenario 4b)

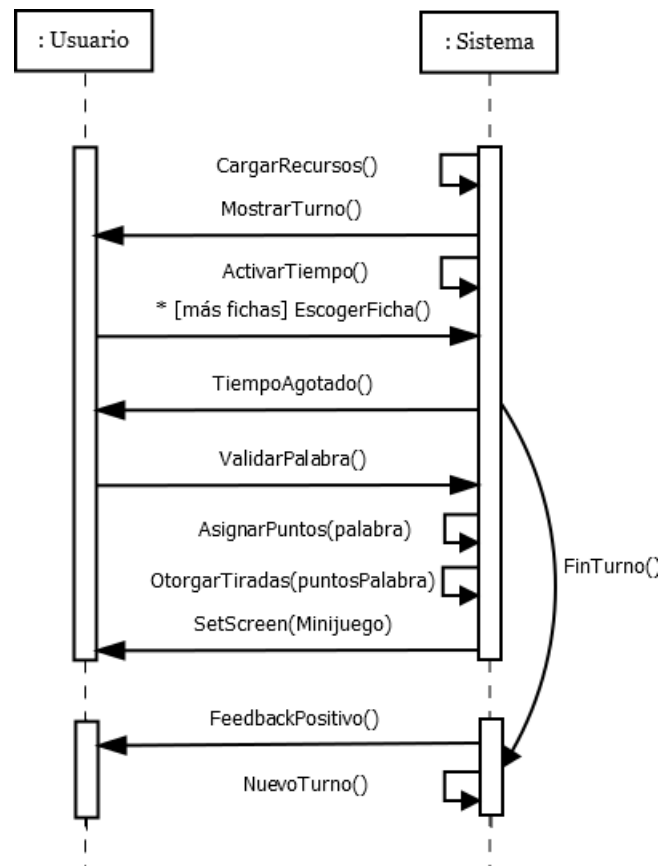


Figura 4.16: Diagrama de secuencia: Formación de palabras (escenario 4b)

Operación: TiempoAgotado()

- **Actores:** Sistema.
- **Responsabilidades:** Notificar que el tiempo se ha agotado.
- **Referencias cruzadas:** Caso de uso: Formación de palabras.
- **Precondiciones:** El usuario ha empezado un juego y el tiempo para la formación de palabras se ha agotado.
- **Postcondiciones:** El sistema notifica al usuario que se ha agotado su tiempo.

Operación: FinTurno()

- **Actores:** Sistema.
- **Responsabilidades:** Terminar el turno actual.
- **Referencias cruzadas:** Caso de uso: Formación de palabras.

- *Precondiciones:* El usuario ha empezado un juego y el tiempo para la formación de palabras se ha agotado.
- *Postcondiciones:* El sistema termina el turno actual.

Caso de uso: Formación de palabras (escenario 5a)

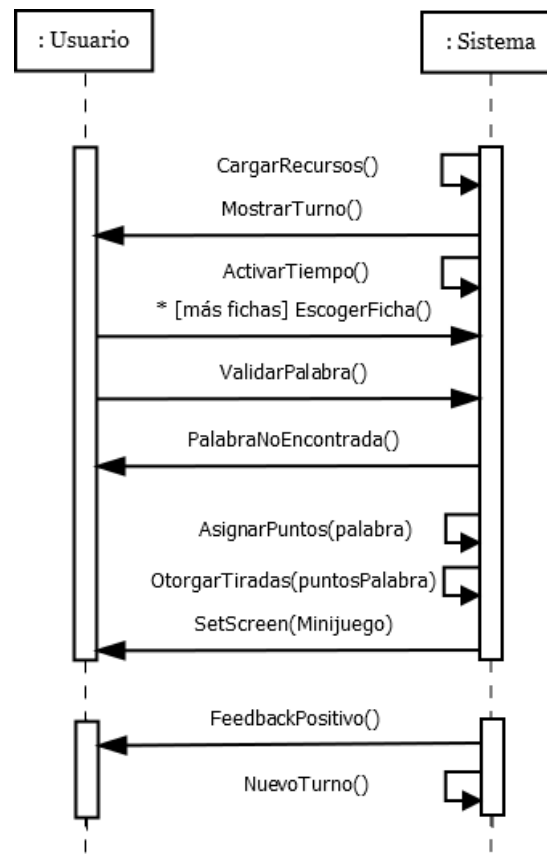


Figura 4.17: Diagrama de secuencia: Formación de palabras (escenario 5a)

Operación: PalabraNoEncontrada()

- *Actores:* Sistema.
- *Responsabilidades:* Mostrar al usuario que la palabra no pertenece a la categoría que se está jugando.
- *Referencias cruzadas:* Caso de uso: Formación de palabras.
- *Precondiciones:* El usuario ha empezado un juego y ha enviado una palabra a validar.
- *Postcondiciones:* El sistema muestra al usuario un mensaje comunicándole que la palabra no se ha encontrado.

Caso de uso: Formación de palabras (escenario 2-4)

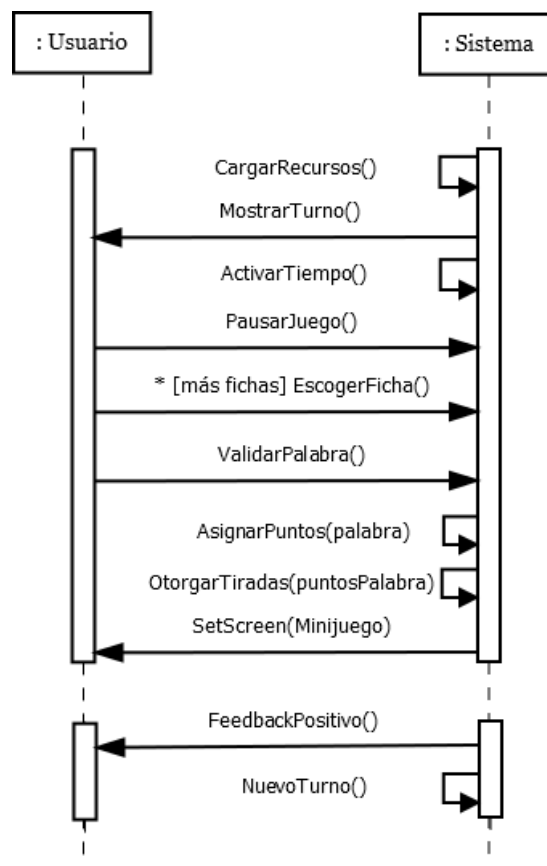


Figura 4.18: Diagrama de secuencia: Formación de palabras (escenario 2-4)

Operación: PausarJuego()

- *Actores:* Usuario.
- *Responsabilidades:* El usuario pausa el juego.
- *Referencias cruzadas:* Caso de uso: Formación de palabras.
- *Precondiciones:* El usuario ha empezado un juego y pulsa el botón de pausar el juego.
- *Postcondiciones:* El juego se pausa.

Caso de uso: Minijuego (escenario principal y 4a)

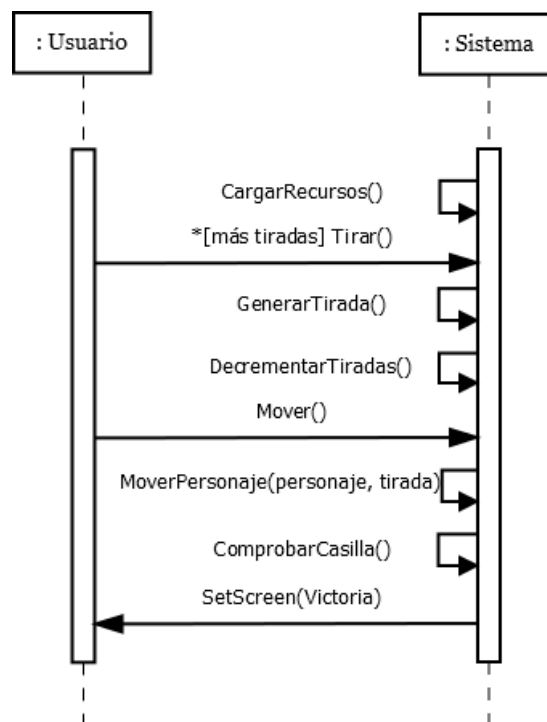


Figura 4.19: Diagrama de secuencia: Minijuego (escenario principal y 4a)

Operación: Tirar()

- *Actores:* Usuario.
- *Responsabilidades:* El usuario tira el dado.
- *Referencias cruzadas:* Caso de uso: Minijuego.
- *Precondiciones:* El usuario se encuentra en el minijuego.
- *Postcondiciones:* Ninguna.

Operación: GenerarTirada()

- *Actores:* Sistema.
- *Responsabilidades:* Generar una tirada aleatoria de un dado de 6 caras.
- *Referencias cruzadas:* Caso de uso: Minijuego.
- *Precondiciones:* El usuario se encuentra en el minijuego y ha tirado.
- *Postcondiciones:* Genera una tirada aleatoria del 1 al 6.

Operación: DecrementarTiradas()

- *Actores:* Sistema.
- *Responsabilidades:* Decrementar en uno las tiradas disponibles.
- *Referencias cruzadas:* Caso de uso: Minijuego.
- *Precondiciones:* El usuario se encuentra en el minijuego y ha tirado.
- *Postcondiciones:* Tiradas = Tiradas - 1;

Operación: Mover()

- *Actores:* Usuario.
- *Responsabilidades:* El usuario mueve a su personaje.
- *Referencias cruzadas:* Caso de uso: Minijuego.
- *Precondiciones:* El usuario se encuentra en el minijuego.
- *Postcondiciones:* Ninguna.

Operación: MoverPersonaje(personaje, tirada)

- *Actores:* Sistema.
- *Responsabilidades:* Mover al personaje por el tablero.
- *Referencias cruzadas:* Caso de uso: Minijuego.
- *Precondiciones:* El usuario se encuentra en el minijuego y ha movido.
- *Postcondiciones:* El sistema mueve al personaje tantas casillas como indica la tirada.

Operación: ComprobarCasilla()

- *Actores:* Sistema.
- *Responsabilidades:* Comprobar el tipo de casilla a donde se ha movido el personaje.
- *Referencias cruzadas:* Caso de uso: Minijuego.
- *Precondiciones:* El usuario se encuentra en el minijuego y ha movido.
- *Postcondiciones:* El sistema comprueba si la casilla es vacía, tiene un agujero o es la meta.

Caso de uso: Minijuego (escenario 4b (2a))

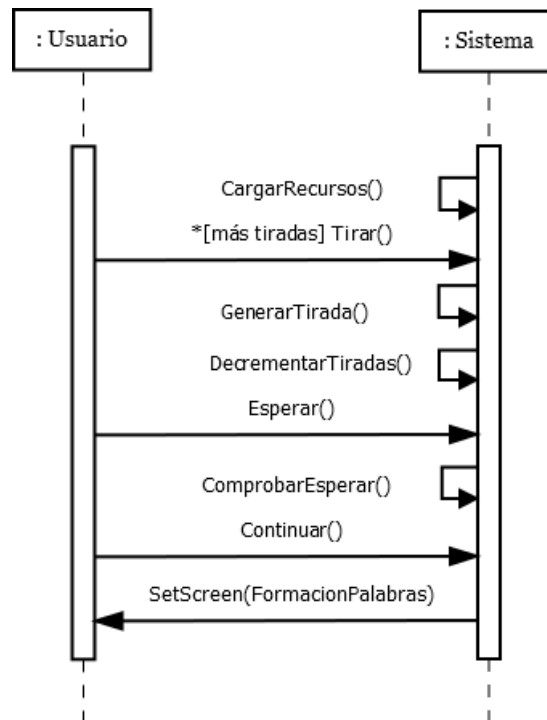


Figura 4.20: Diagrama de secuencia: Minijuego (escenario 4b (2a))

Operación: Esperar()

- *Actores:* Usuario.
- *Responsabilidades:* El usuario espera al siguiente turno.
- *Referencias cruzadas:* Caso de uso: Minijuego.
- *Precondiciones:* El usuario se encuentra en el minijuego y decide esperar al siguiente turno.
- *Postcondiciones:* Ninguna.

Operación: ComprobarEsperar()

- *Actores:* Sistema.
- *Responsabilidades:* Comprobar si se puede usar la función esperar.
- *Referencias cruzadas:* Caso de uso: Minijuego.
- *Precondiciones:* El usuario se encuentra en el minijuego y pulsa el botón esperar.
- *Postcondiciones:* El sistema comprueba si el usuario puede hacer uso de la función esperar.

Operación: Continuar()

- *Actores:* Usuario.
- *Responsabilidades:* El usuario finaliza el minijuego.
- *Referencias cruzadas:* Caso de uso: Minijuego.
- *Precondiciones:* El usuario se encuentra en el minijuego y ha esperado o terminado de jugar sus tiradas.
- *Postcondiciones:* Ninguna.

Caso de uso: Minijuego (escenario 4b (2b))

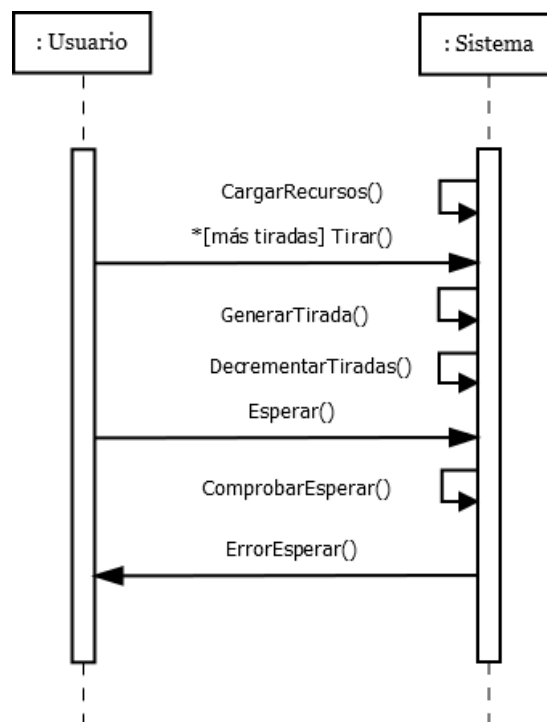


Figura 4.21: Diagrama de secuencia: Minijuego (escenario 4b (2b))

Operación: ErrorEsperar()

- *Actores:* Sistema.
- *Responsabilidades:* Indicar al usuario que no puede esperar.
- *Referencias cruzadas:* Caso de uso: Minijuego.
- *Precondiciones:* El usuario se encuentra en el minijuego y ha usado la función esperar.
- *Postcondiciones:* El sistema muestra un mensaje diciéndole que no puede usar esperar hasta que no mueva al menos una de sus tiradas.

Caso de uso: Minijuego (escenario 6c)

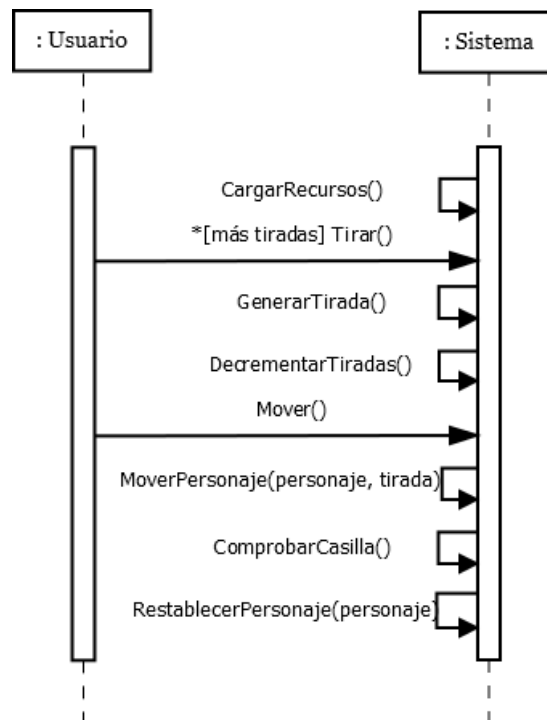


Figura 4.22: Diagrama de secuencia: Minijuego (escenario 6c)

Operación: RestablecerPersonaje(personaje)

- *Actores:* Sistema.
- *Responsabilidades:* Devolver al personaje a su casilla de salida correspondiente.
- *Referencias cruzadas:* Caso de uso: Minijuego.
- *Precondiciones:* El usuario se encuentra en el minijuego y su personaje ha caído en un agujero.
- *Postcondiciones:* El sistema traslada al personaje a la casilla de salida.

Caso de uso: UsarPista (escenario principal)

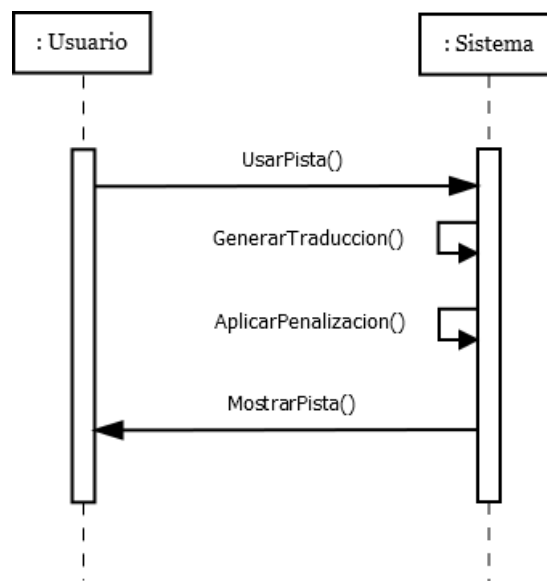


Figura 4.23: Diagrama de secuencia: Usar Pista (escenario principal)

Operación: GenerarTraduccion()

- *Actores:* Sistema.
- *Responsabilidades:* Buscar traducción al castellano de una palabra del tablero.
- *Referencias cruzadas:* Caso de uso: UsarPista.
- *Precondiciones:* El usuario se encuentra en la formación de palabras y ha usado el botón de pista traducción.
- *Postcondiciones:* El sistema genera la traducción.

Operación: AplicarPenalizacion()

- *Actores:* Sistema.
- *Responsabilidades:* Penalizar al jugador por usar una pista.
- *Referencias cruzadas:* Caso de uso: UsarPista.
- *Precondiciones:* El usuario se encuentra en la formación de palabras y ha usado una pista.
- *Postcondiciones:* El sistema aplica la penalización correspondiente por la pista.

Caso de uso: UsarPista (escenario 1a)

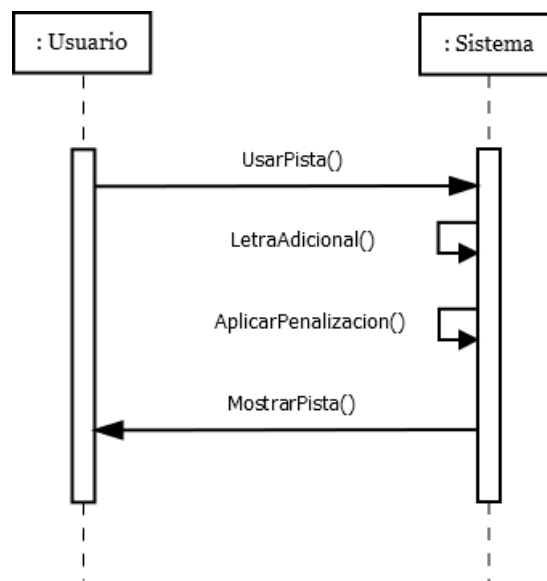


Figura 4.24: Diagrama de secuencia: Usar Pista (escenario 1a)

Operación: LetraAdicional()

- *Actores:* Sistema.
- *Responsabilidades:* Dar la primera letra de una de las palabras del tablero.
- *Referencias cruzadas:* Caso de uso: UsarPista.
- *Precondiciones:* El usuario se encuentra en la formación de palabras y ha usado una pista.
- *Postcondiciones:* El sistema devuelve la primera letra.

Caso de uso: UsarPista (escenario 1b)

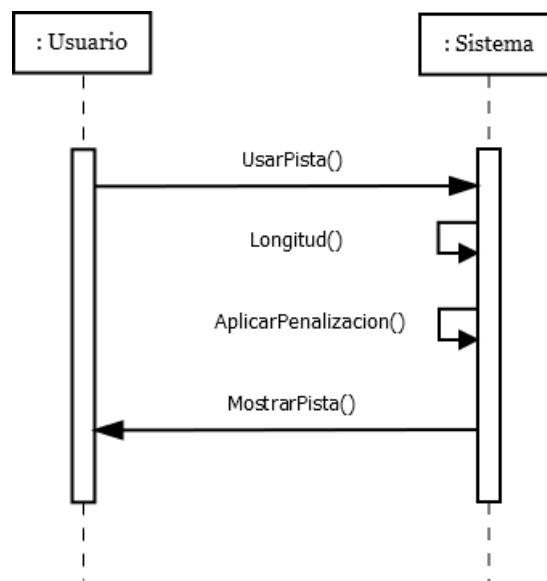


Figura 4.25: Diagrama de secuencia: Usar Pista (escenario 1b)

Operación: Longitud()

- *Actores:* Sistema.
- *Responsabilidades:* Dar la longitud de una de las palabras del tablero.
- *Referencias cruzadas:* Caso de uso: UsarPista.
- *Precondiciones:* El usuario se encuentra en la formación de palabras y ha usado una pista.
- *Postcondiciones:* El sistema devuelve la longitud.

Caso de uso: PausarJuego (escenario principal)

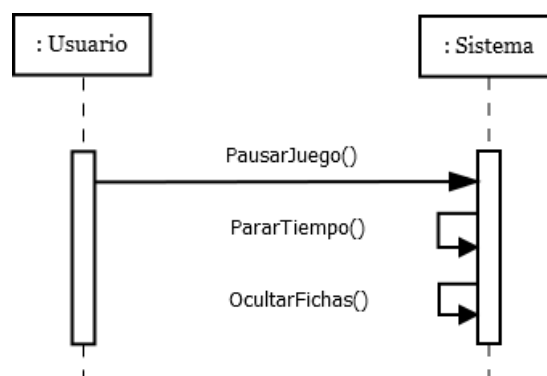


Figura 4.26: Diagrama de secuencia: Pausar Juego (escenario principal)

Operación: PararTiempo()

- *Actores:* Sistema.
- *Responsabilidades:* Para el tiempo de juego.
- *Referencias cruzadas:* Caso de uso: PausarJuego.
- *Precondiciones:* El usuario se encuentra en la formación de palabras y pausa el juego.
- *Postcondiciones:* El sistema pausa el tiempo de juego.

Operación: OcultarFichas()

- *Actores:* Sistema.
- *Responsabilidades:* Ocultar las fichas de juego.
- *Referencias cruzadas:* Caso de uso: PausarJuego.
- *Precondiciones:* El usuario se encuentra en la formación de palabras y pausa el juego.
- *Postcondiciones:* El sistema oculta las fichas del juego.

Caso de uso: PausarJuego (escenario 3a)

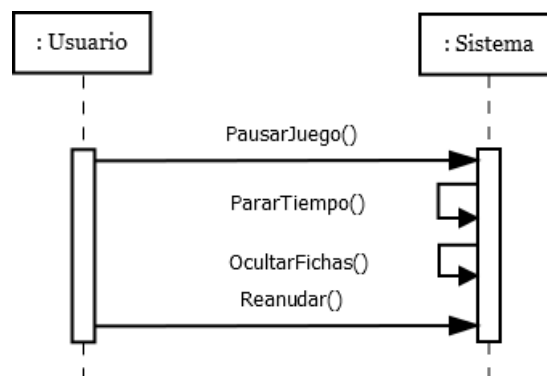


Figura 4.27: Diagrama de secuencia: Pausar Juego (escenario 3a)

Operación: Reanudar()

- *Actores:* Usuario.
- *Responsabilidades:* El usuario reanuda el juego.
- *Referencias cruzadas:* Caso de uso: PausarJuego.
- *Precondiciones:* El usuario se encuentra en la formación de palabras y tenía pausado el juego.
- *Postcondiciones:* Ninguna.

Caso de uso: PausarJuego (escenario 3b)

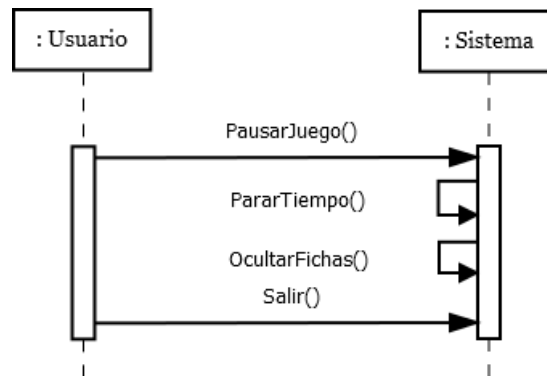


Figura 4.28: Diagrama de secuencia: Pausar Juego (escenario 3b)

Operación: Salir()

- *Actores:* Usuario.
- *Responsabilidades:* El usuario sale del juego.
- *Referencias cruzadas:* Caso de uso: PausarJuego.
- *Precondiciones:* El usuario se encuentra en la formación de palabras y tenía pausado el juego.
- *Postcondiciones:* Ninguna.

Caso de uso: Reanudar

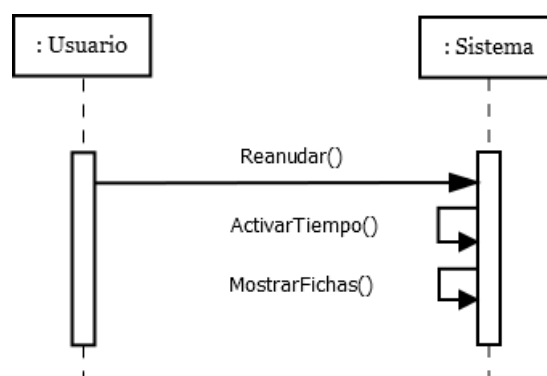


Figura 4.29: Diagrama de secuencia: Reanudar

Operación: MostrarFichas()

- *Actores:* Sistema.

- *Responsabilidades*: Mostrar las fichas de juego.
- *Referencias cruzadas*: Caso de uso: Reanudar.
- *Precondiciones*: El usuario reanuda el juego.
- *Postcondiciones*: El sistema muestra las fichas de juego.

Caso de uso: Salir

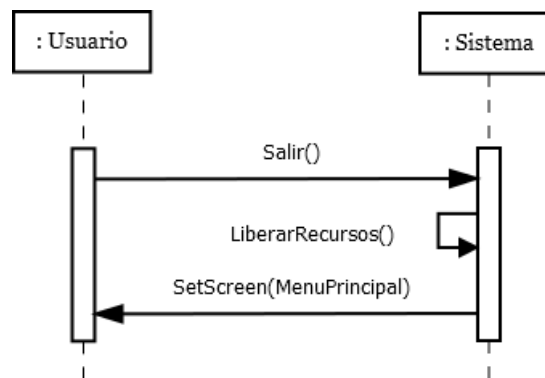


Figura 4.30: Diagrama de secuencia: Salir

Operación: LiberarRecursos()

- *Actores*: Sistema.
- *Responsabilidades*: Liberar recursos de las pantallas de juego.
- *Referencias cruzadas*: Caso de uso: Salir.
- *Precondiciones*: El usuario tenía pausado del juego.
- *Postcondiciones*: El sistema libera los recursos usados en las pantallas de juego.

Caso de uso: Estadísticas (escenario principal)

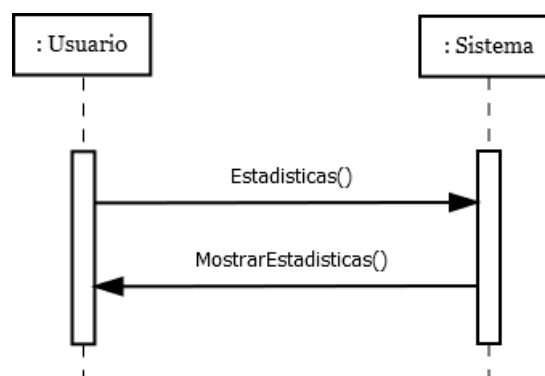


Figura 4.31: Diagrama de secuencia: Estadísticas (escenario principal)

Operación: MostrarEstadisticas()

- *Actores:* Sistema.
- *Responsabilidades:* Mostrar las estadísticas generales.
- *Referencias cruzadas:* Caso de uso: Estadísticas.
- *Precondiciones:* El usuario ha pulsado el botón estadísticas del menú principal.
- *Postcondiciones:* El sistema muestra las estadísticas generales del jugador.

Caso de uso: Estadísticas (escenario 3a)

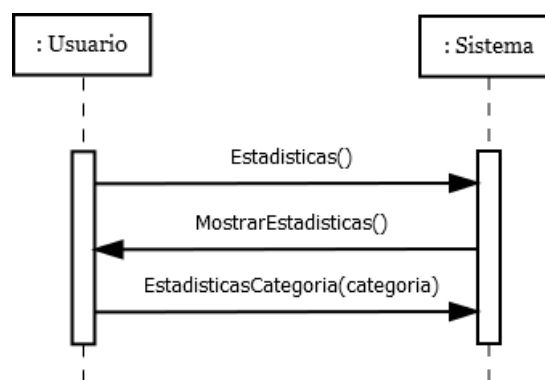


Figura 4.32: Diagrama de secuencia: Estadísticas (escenario 3a)

Operación: EstadisticasCategoria(categoria)

- *Actores:* Usuario.
- *Responsabilidades:* El usuario quiere ver las estadísticas particulares de una categoría.
- *Referencias cruzadas:* Caso de uso: Estadísticas.
- *Precondiciones:* El usuario se encuentra en la pantalla de estadísticas y pulsa el botón de una categoría.
- *Postcondiciones:* Ninguna.

Caso de uso: Mostrar Estadísticas Categoría X

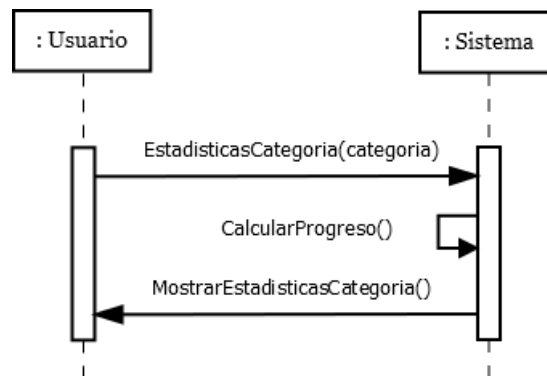


Figura 4.33: Diagrama de secuencia: Estadísticas Categoría X

Operación: CalcularProgreso()

- *Actores:* Sistema.
- *Responsabilidades:* Calcular el progreso de la categoría.
- *Referencias cruzadas:* Caso de uso: MostrarEstadisticasCategoriaX
- *Precondiciones:* El usuario se encuentra en la pantalla de estadísticas generales y quiere ver una categoría en particular.
- *Postcondiciones:* El sistema calcula el progreso.

Operación: MostrarEstadisticasCategoria()

- *Actores:* Sistema.
- *Responsabilidades:* Mostrar las estadísticas de la categoría.
- *Referencias cruzadas:* Caso de uso: MostrarEstadisticasCategoriaX
- *Precondiciones:* El usuario se encuentra en la pantalla de estadísticas generales y quiere ver una categoría en particular.
- *Postcondiciones:* El sistema muestra la información.

Caso de uso: Ajustes (escenario principal)

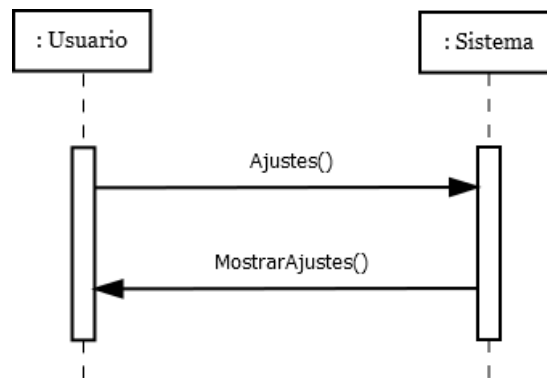


Figura 4.34: Diagrama de secuencia: Ajustes (escenario principal)

Operación: `MostrarAjustes()`

- *Actores:* Sistema.
- *Responsabilidades:* Mostrar los ajustes.
- *Referencias cruzadas:* Caso de uso: Ajustes.
- *Precondiciones:* El usuario se encuentra en el menú principal y pulsa ajustes.
- *Postcondiciones:* El sistema muestra la pantalla de ajustes.

Caso de uso: Ajustes (escenario 3a)

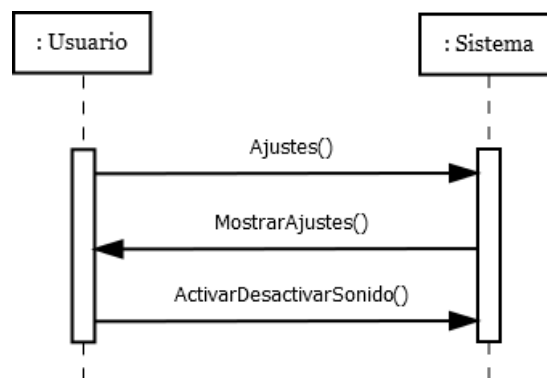


Figura 4.35: Diagrama de secuencia: Ajustes (escenario 3a)

Operación: ActivarDesactivarSonido()

- *Actores:* Usuario.
- *Responsabilidades:* El usuario cambia los ajustes de sonido.
- *Referencias cruzadas:* Caso de uso: Ajustes.
- *Precondiciones:* El usuario se encuentra en la pantalla de ajustes.
- *Postcondiciones:* Ninguna.

Caso de uso: Ajustes (escenario 3b)

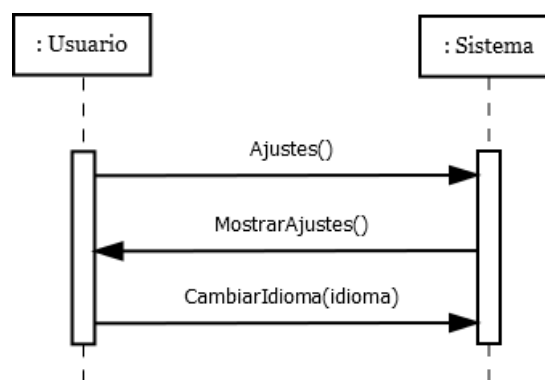


Figura 4.36: Diagrama de secuencia: Ajustes (escenario 3b)

Operación: CambiarIdioma(idioma)

- *Actores:* Usuario.
- *Responsabilidades:* El usuario cambia el idioma.
- *Referencias cruzadas:* Caso de uso: Ajustes.
- *Precondiciones:* El usuario se encuentra en la pantalla de ajustes.
- *Postcondiciones:* Ninguna.

Caso de uso: Activar/Desactivar Sonido

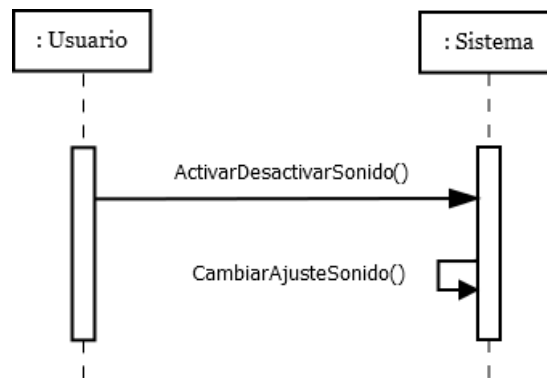


Figura 4.37: Diagrama de secuencia: Activar/Desactivar Sonido

Operación: CambiarAjusteSonido()

- *Actores:* Sistema.
- *Responsabilidades:* Activar o desactivar el sonido.
- *Referencias cruzadas:* Caso de uso: Activar/Desactivar Sonido.
- *Precondiciones:* El usuario se encuentra en la pantalla de ajustes y quiere activar/desactivar el sonido.
- *Postcondiciones:* El sistema activa o desactiva el sonido en función del ajuste.

Caso de uso: Cambiar Idioma

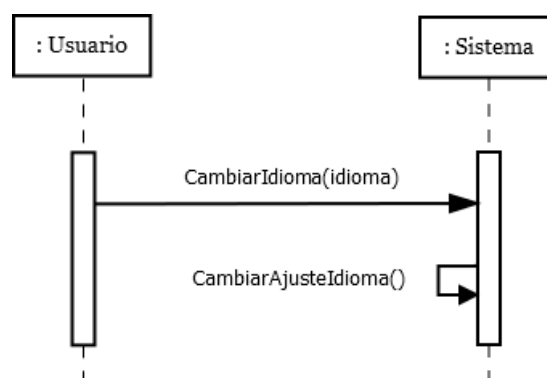


Figura 4.38: Diagrama de secuencia: Cambiar idioma

Operación: CambiarAjusteIdioma()

- *Actores:* Sistema.
- *Responsabilidades:* Cambiar el idioma del juego.
- *Referencias cruzadas:* Caso de uso: Cambiar Idioma.
- *Precondiciones:* El usuario se encuentra en la pantalla de ajustes y quiere cambiar el idioma de juego
- *Postcondiciones:* El sistema cambia al idioma de juego seleccionado por el usuario.

Caso de uso: Instrucciones

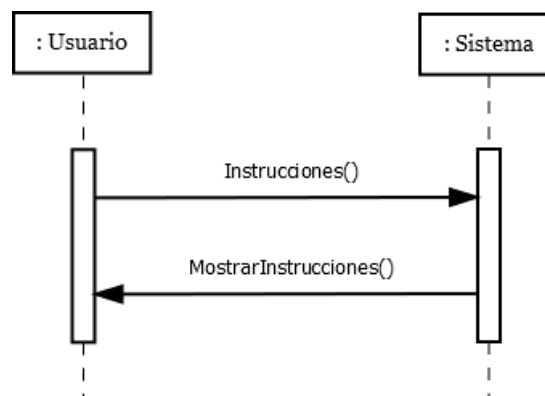


Figura 4.39: Diagrama de secuencia: Instrucciones

Operación: MostrarInstrucciones()

- *Actores:* Sistema.
- *Responsabilidades:* Mostrar las instrucciones.
- *Referencias cruzadas:* Caso de uso: Instrucciones.
- *Precondiciones:* El usuario se encuentra en el menú principal y ha pulsado instrucciones.
- *Postcondiciones:* El sistema muestra la información de las instrucciones.

Caso de uso: Salir Aplicación

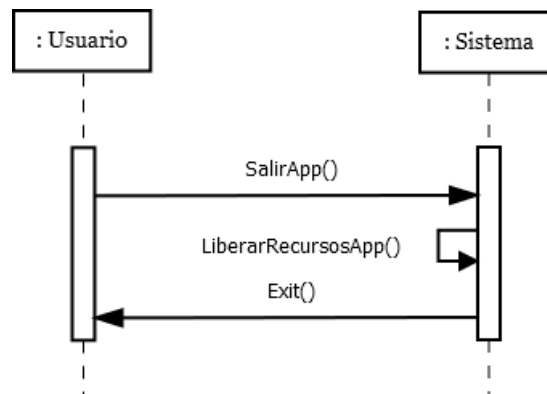


Figura 4.40: Diagrama de secuencia: Salir aplicación

Operación: LiberarRecursosApp()

- *Actores:* Sistema.
- *Responsabilidades:* Liberar todos los recursos que compone la APP.
- *Referencias cruzadas:* Caso de uso: Salir aplicación.
- *Precondiciones:* El usuario se encuentra en el menú principal y pulsa salir.
- *Postcondiciones:* El sistema libera toda la memoria usada por la aplicación.

Operación: Exit()

- *Actores:* Sistema.
- *Responsabilidades:* Salir de la aplicación.
- *Referencias cruzadas:* Caso de uso: Salir aplicación.
- *Precondiciones:* El usuario se encuentra en el menú principal y pulsa salir.
- *Postcondiciones:* El sistema termina la aplicación.

Capítulo 5

Diseño del sistema

Este capítulo comprende todo el diseño del software. Recorreremos la interfaz gráfica completa y mostraremos imágenes de las pantallas que la componen para un mejor entendimiento del análisis hecho en el capítulo anterior. Por supuesto, gracias a la fase anterior, interpretaremos mejor el diseño de la aplicación.

5.1. Elementos gráficos

Los elementos gráficos constituyen principalmente la mayor fuente de recursos para cualquier aplicación que requiera de una interfaz agradable. Todos los componentes representativos del videojuego han sido creados por el autor mediante el uso de herramientas gráficas, o bien han sido tomados/editados de la página de recursos libres *OpenGameArt* ¹ bajo licencia gratuita y de uso comercial.

5.2. Elementos de sonido

Puesto que no poseía los conocimientos ni herramientas necesarias para producir efectos de sonido, estos han sido tomados de sitios web de recursos gratuitos como *FreeSound* ² o *Sound-Bible* ³.

5.3. Interfaz gráfica

Para unir los elementos anteriores necesitamos crear una interfaz gráfica simple y que a la vez sea elegante. Sobre todo, lo que se tendrá más en cuenta es que sea totalmente intuitiva y no suponga problemas para el usuario.

Seguidamente, mostramos como se navega entre las distintas pantallas del juego:

¹<http://opengameart.org>

²<http://www.freesound.org>

³<http://www.soundbible.com>

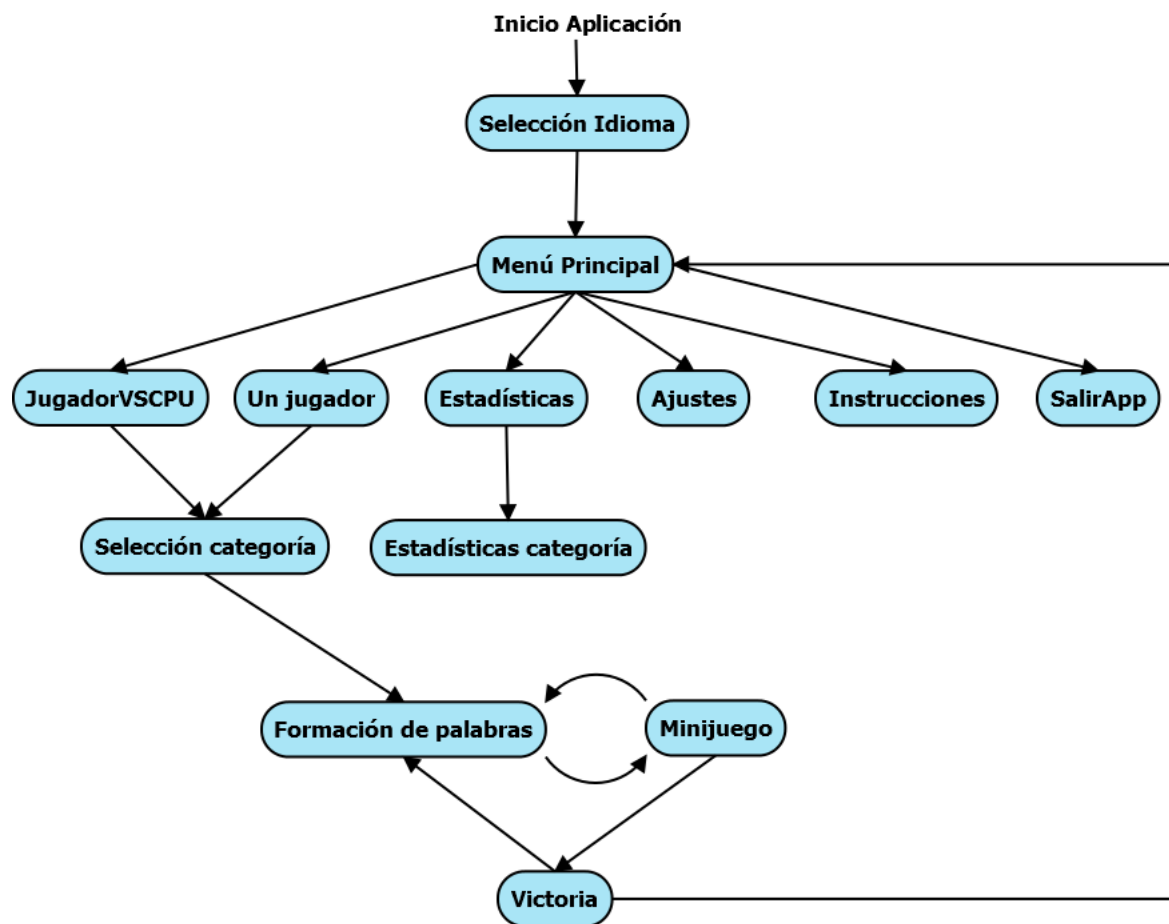


Figura 5.1: Navegación de pantallas

Y las respectivas pantallas son las siguientes:



Figura 5.2: Pantalla de selección de idioma

Esta pantalla es la que inicia la aplicación, tiene dos imágenes que se pueden pulsar y que representan las banderas de los idiomas disponibles para el juego.

Aunque después se puede cambiar el idioma en la pantalla de ajustes (figura 5.7), al principio hay que escoger el idioma inicial del juego.



Figura 5.3: Pantalla de Menú principal

La pantalla del menú principal es como el punto de encuentro de la aplicación. Desde aquí podemos acceder a todas las opciones que dispone el juego.

La pantalla está compuesta por una etiqueta que muestra el título del juego y seis botones con las elecciones que podemos hacer. A continuación exploraremos todas las pantallas a las que nos llevan estos botones.



Figura 5.4: Pantalla de Selección de categoría

Tanto si pulsamos en el modo Jugador VS CPU como en Un jugador se nos dirigirá a la pantalla de selección de categoría previa a empezar el juego. Aquí escogemos la categoría para la partida.

En cuanto a los elementos, está compuesta de una etiqueta de título de pantalla, un botón por cada categoría de juego y un botón para volver atrás por si reconsideramos no jugar.



Figura 5.5: Pantalla de Estadísticas

Como tercera opción del menú principal tenemos las estadísticas del juego. Al pulsarlo nos muestra las categorías generales y la posibilidad de ver la información de una categoría en particular.

Los elementos generales comprenden: ratio de acierto, juegos ganados, juegos perdidos, palabra más larga, mejor palabra hablando en puntos y pistas usadas. A continuación veremos como se muestran las estadísticas para una categoría en particular.

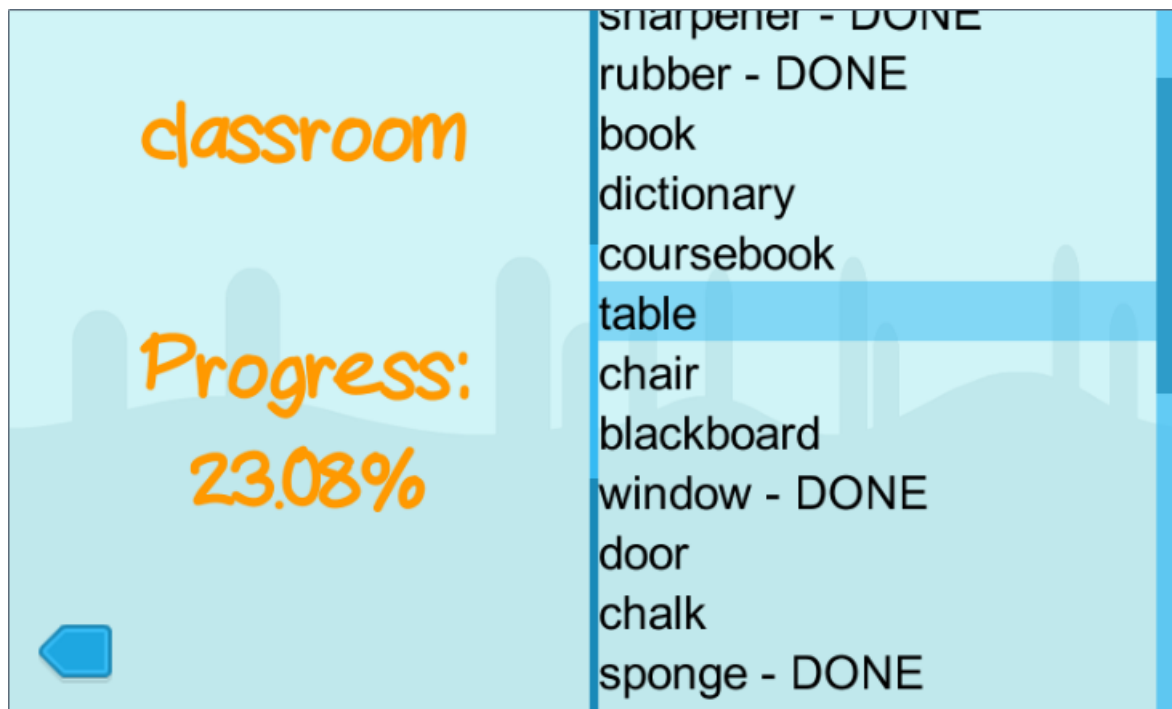


Figura 5.6: Pantalla de Estadísticas por categoría

Pulsando en una de las categorías de la pantalla anterior, el sistema nos carga la información relativa a una categoría individual.

En esta pantalla nos encontramos con los siguientes elementos: una etiqueta con el título de la categoría escogida, el progreso que llevamos en la categoría y una lista con scroll vertical que nos revela las palabras que hay en la categoría. Además si hemos formado alguna, la palabra vendrá acompañada de la cadena "DONE", lo cual significa que la hemos hecho en alguna de las partidas.



Figura 5.7: Pantalla de Ajustes

La cuarta opción del menú principal nos muestra los ajustes de la aplicación. En ella podemos activar/desactivar el sonido o cambiar el idioma. También podemos ver información sobre los autores del software, así como su versión y repositorio.

Entre sus elementos, tenemos una etiqueta como viene siendo costumbre con el título de la pantalla donde estamos ubicados, un checkbox para activar o desactivar el sonido y las dos banderas para cambiar de idioma. El idioma seleccionado actualmente viene indicado mediante una marca en la parte inferior derecha de la bandera en cuestión.

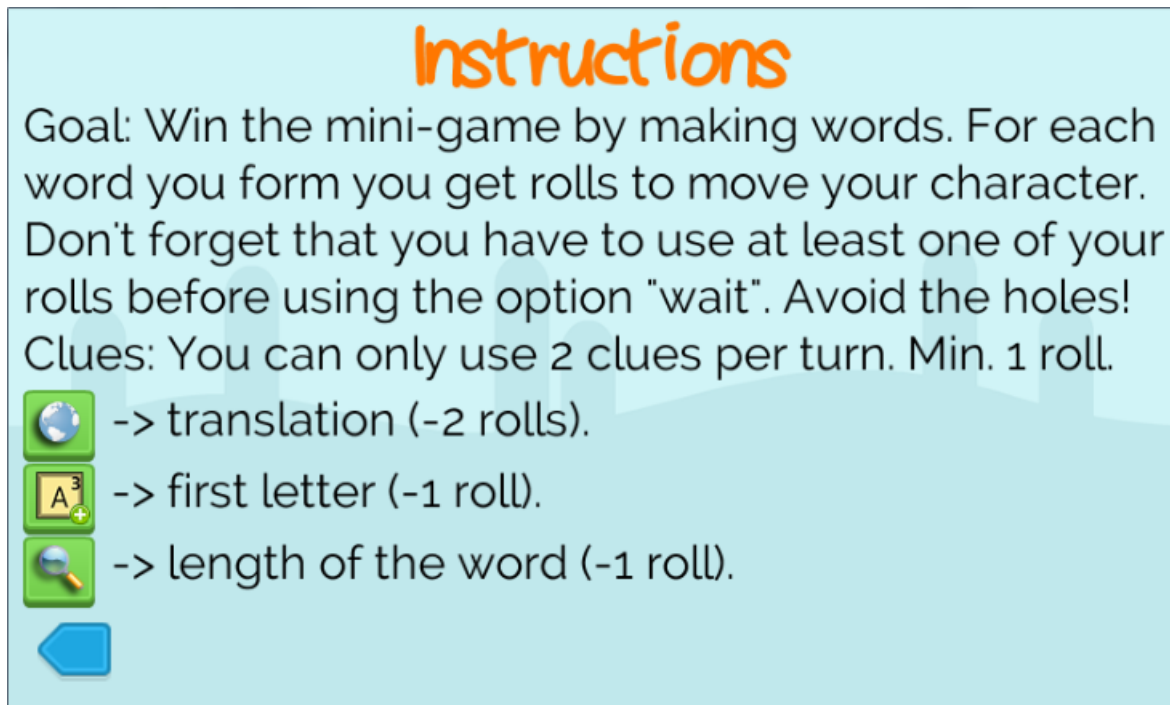


Figura 5.8: Pantalla de Instrucciones

La sexta opción es la pantalla de instrucciones. Esta pantalla tiene una breve explicación del objetivo del juego así como la descripción de cada una de las pistas.

Por supuesto esta pantalla, no desvela los entresijos del juego. Para complementar esta pantalla tenemos un manual de usuario incluido en esta memoria como apéndice.

Esta pantalla se compone básicamente de etiquetas de texto y imágenes que representan los botones de las pistas para una mejor comprensión.

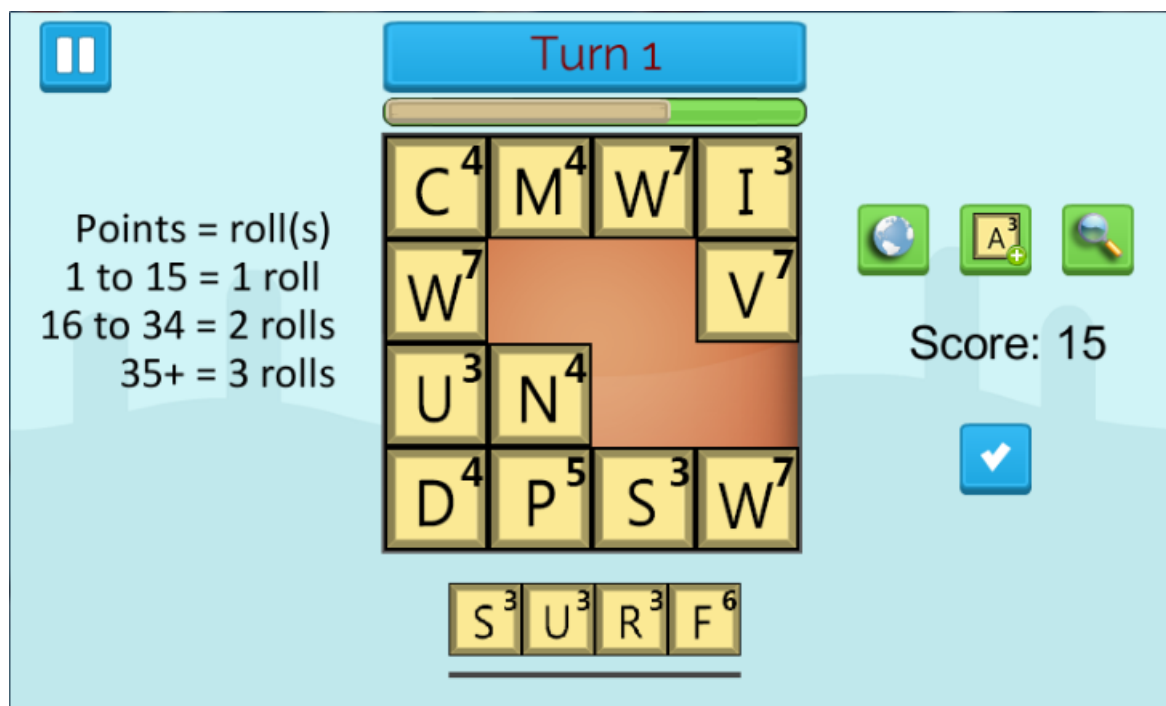


Figura 5.9: Pantalla de juego: Formación de palabras

Pasamos pues al juego después de haber escogido una categoría en la pantalla de selección de categoría (figura 5.4).

En esta vista, tenemos la parte de formación de palabras de nuestro juego. Aquí es donde crearemos palabras con la ayuda de las fichas que tenemos disponibles.

Sus elementos son los siguientes: botón de pausa, etiqueta con información de puntos y tiradas, etiqueta de turno, barra de tiempo, botones de pistas, etiqueta de puntuación y botón de validación de palabras.

Destacamos que cuando se pulsa el botón de pausa se abre el menú contextual de la figura 5.10.

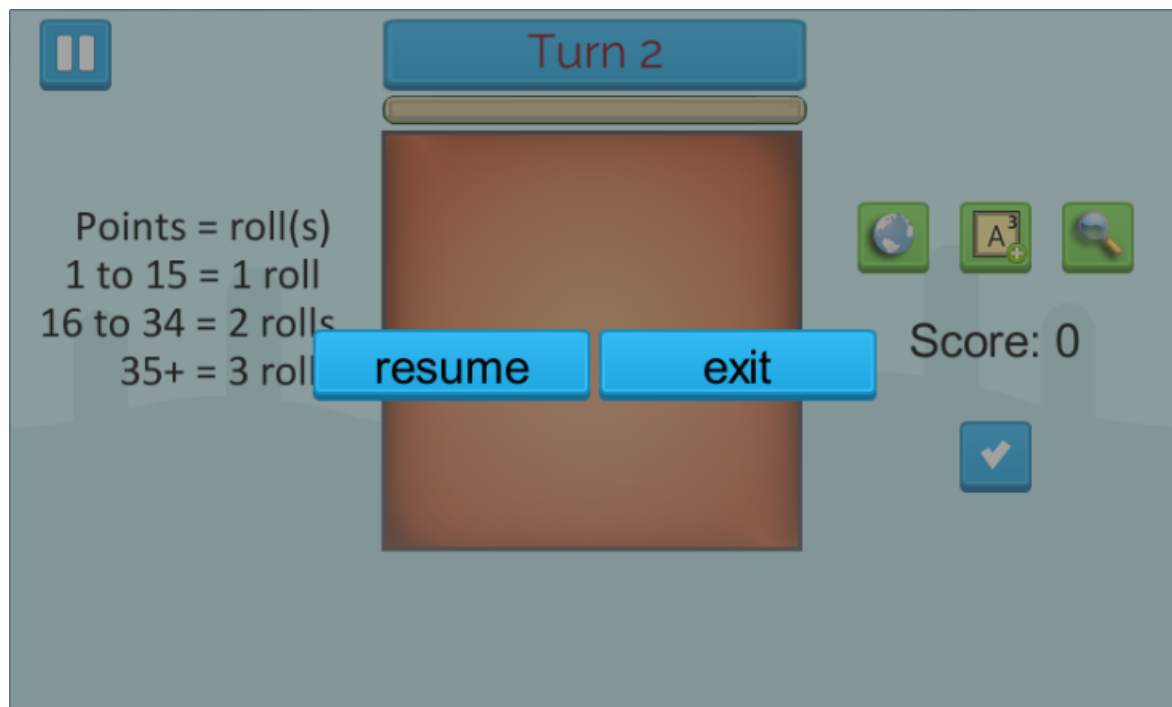


Figura 5.10: Pantalla de juego: Formación de palabras en pausa

Tal como mencionamos anteriormente, pausando el juego se abre este pequeño menú con dos opciones: resume y exit.

Con el botón resume volvemos al juego y con el botón exit salimos al menú principal.

Cuando se accede a esta pantalla se ocultan las letras para evitar que se aproveche la situación para hacer trampas.

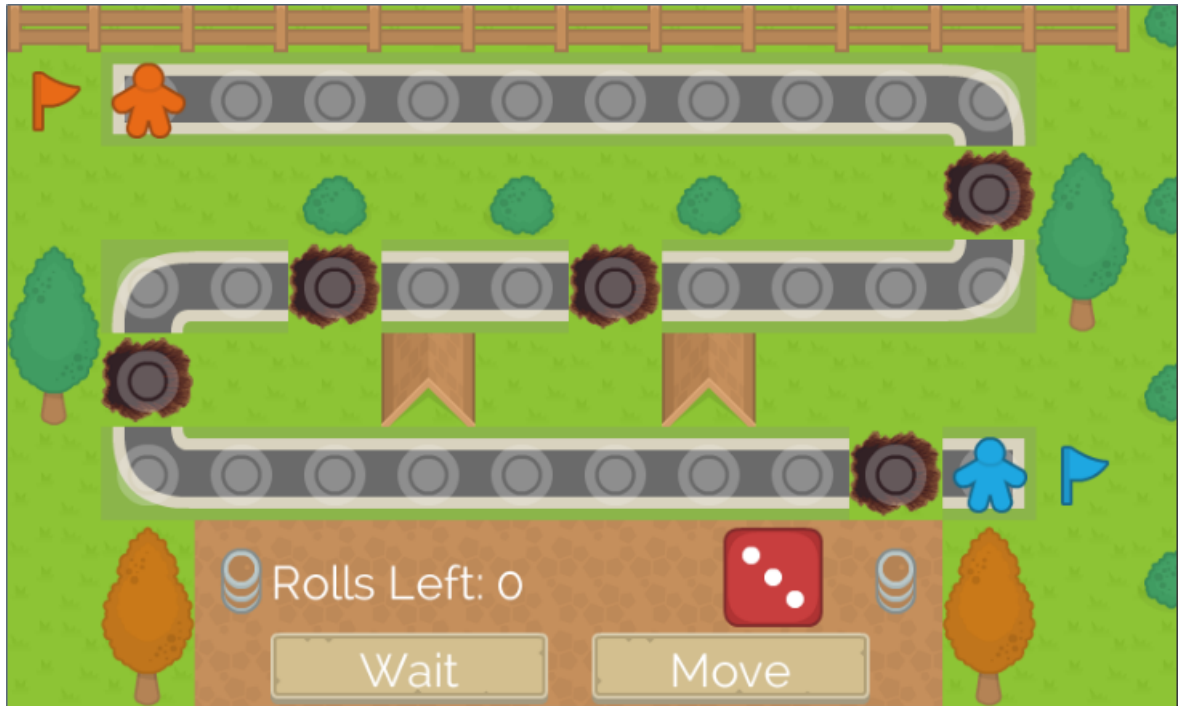


Figura 5.11: Pantalla de juego: Minijuego

Una vez que validemos una palabra, se nos llevará a la pantalla del minijuego donde gastaremos nuestras tiradas.

Los elementos que encontramos son los siguientes: tablero del juego, fichas de personajes, etiqueta de tiradas restantes, dado y botones de Tirar/esperar y mover.

La iteración del juego alternará entre esta pantalla y la de formación de palabras (figura 5.9) hasta que uno de los jugadores (o jugador en el modo Un jugador) alcance la otra punta del tablero.



Figura 5.12: Pantalla de Victoria

Una vez se alcanza el objetivo del juego, se nos redirige a la pantalla de victoria donde tenemos un breve resumen de la partida.

Los elementos que nos encontramos son: etiquetas de resumen (turnos y palabras totales), etiqueta del ganador, palabras que hemos hecho frente a las que hay en la categoría dispuesta en un scroll vertical y dos botones (jugar otra vez y salir).

5.4. Arquitectura general del sistema

Como hemos venido mencionando, nuestro juego hace uso de la librería Scene2D para el diseño del juego. La razón de ello, es porque esta API nos permite añadir elementos de forma fácil y gestionarlos mediante métodos prácticamente necesarios que ya vienen implementados. Además, está enfocada tanto a la creación de elementos en 2D como al diseño de la interfaz de usuario.

Scene2D conforma una estructura jerárquica de actores que se añaden a un escenario, es decir, tiene una estructura de árbol pero con una jerarquía definida. Esto significa, que los actores están más arriba del nodo raíz, que es el escenario, se van a mostrar encima de los que están debajo en el caso de que ocupen la misma posición.

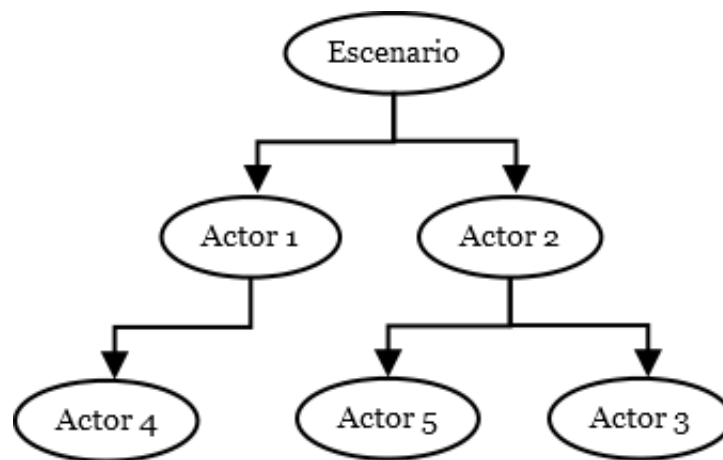


Figura 5.13: Estructura de Scene2D

5.4.1. Escenario

El escenario es donde ocurren todos los sucesos, se disponen los actores, se controlan los eventos y se ejecutan las acciones.

Principalmente tiene dos métodos importantes: `act()` y `draw()`. Todos los actores cuentan con dichos métodos que pueden ser sobrecargados para dar un comportamiento distinto. El primero indica como tienen que actuar los actores en el escenario, es decir, que acciones realizan, como las llevan a cabo, etc.

El segundo, da información sobre como han de ser representados los actores, es decir, como han de dibujarse en el escenario, en que posición, etc.

Cuando se han definido estos métodos para los actores, se llama a los métodos del escenario que llaman recursivamente a todos los métodos `act()` y `draw()` de los actores encapsulando que hacen cada uno de ellos en sus clases.

Nuestro juego tiene para cada pantalla un escenario. Todas las pantallas contienen información base que se comparte. Cada pantalla es una clase que desciende a su vez de otra clase llamada

pantalla base. En ella se define el comportamiento común de todas las pantallas. Las clases para la realización de esta implementación son las siguientes:

BaseScreen: Clase abstracta que engloba todos los aspectos comunes de las pantallas. Como el escenario por ejemplo.

LanguageSelectionScreen: Pantalla encargada de mostrar los idiomas disponibles de la aplicación.

MenuScreen: Pantalla del menú principal.

CategorySelectionScreen: Pantalla de selección de categoría previa a la partida.

StatsScreen: Pantalla encargada de mostrar las estadísticas.

CategoryStatsScreen: Pantalla encargada de mostrar las estadísticas particulares de cada categoría.

SettingsScreen: Pantalla encargada de mostrar los ajustes del juego.

InstructionsScreen: Pantalla encargada de mostrar las instrucciones.

GameScreen: Pantalla de formación de palabras.

MiniGameScreen: Pantalla del minijuego.

VictoryScreen: Pantalla final de una partida.

5.4.2. Actores

Los actores representan los elementos en los escenarios, desde botones, fichas, paneles, etc. Cada uno de los actores se representan mediante una clase distinta.

La jerarquía de los actores va en función de cuando los definamos, por decirlo de alguna manera, van en el orden de declaración. Por lo que por ejemplo, el actor de más abajo es el fondo de pantalla. El comportamiento y como se dibujan los actores van encapsulados dentro de la clase liberando así el método `render()` de las pantallas. Los clases de actores más importantes del juego son las siguientes:

Button: Actor que representa un botón en el escenario.

Label: Etiqueta que con texto.

Los menús de la aplicación contienen básicamente etiquetas y botones. Ahora veremos los actores relevantes de las pantallas de juego:

Tile: Actor que representa una ficha de letra.

TileBox: Actor que representa el panel donde descansan las fichas de juego (Tiles).

TimeBar: Actor que representa la barra de turno.

TurnBar: Actor que controla y muestra los turnos.

WordScore: Actor que controla la puntuación de la palabra que se está formando.

Dice: Actor que representa el dado en el minijuego.

PauseDialog: Actor que representa el menú contextual de pausa del juego.

Slider: Actor que representa la barra horizontal donde descansan las letras al bajarlas del Tile-Box.

InfoRoll: Actor que representa los umbrales de puntos y tiradas.

RollsLeft: Actor etiqueta que muestra las tiradas que nos quedan en el minijuego.

También hay otros actores que aunque no se representen gráficamente, actúan con el tiempo. Estos son por ejemplo:

HumanPlayer: Actor que representa al usuario de la aplicación.

NPCPlayer: Actor que representa al jugador controlado por la IA.

TurnControl: Actor que controla qué jugador juega en un turno u otro.

Capítulo 6

Implementación

Como bien sabemos, la implementación del software es la fase fundamental del desarrollo, es decir, es donde enfocamos nuestro mayor esfuerzo en convertir las ideas en algo verdaderamente útil.

En este capítulo vamos a ver las características más importantes del proyecto, los principales problemas a los que nos hemos enfrentado y las soluciones que se han seguido.

6.1. Desarrollo

A continuación detallamos el itinerario de implementación de los módulos principales del juego.

6.1.1. Desarrollo del sistema de palabras

Esta etapa del desarrollo fue una de las más costosas, porque constituía el sistema base de palabras, es decir, comprobar que existían y pertenecían a las categorías establecidas. Adicionalmente, el sistema debía englobar y tratar las palabras de igual manera aunque pertenecieran a idiomas distintos. Esto suponía enfrentarse a la gramática de la lengua en cuestión, con sus características y distinciones propias. Que ambas pertenecieran al alfabeto latino, hizo que todo se llevara más cómodamente.

6.1.2. Desarrollo de la lógica del juego

Ahora que teníamos diseñado el sistema de palabras, había que establecer la lógica del juego. Al principio fue simple, el jugador escogía las fichas del tablero y formaba una palabra, el sistema la aceptaba o rechazaba en función de si existía y pertenecía a la categoría. No pretendía nada más, solo probar que se conjuntaba con el sistema de palabras. Acto seguido se añadieron los idiomas restantes, la inteligencia artificial, las pistas y algunos menús para navegar por la aplicación.

Para comprobar que las partidas finalizaban implementé un pequeño sistema de puntuación (que luego se conjuntaría con el minijuego), con tal de establecer el fin del juego y un ganador. Poco a poco iba tomando forma, era ya un juego básico de composición de palabras.

6.1.3. Desarrollo del minijuego

Ahora llegaba la parte interesante a la vez que difícil. Diseñé un tablero para el juego, en el que solo había que ir tirando un dado y moviendo tu personaje hasta llegar a la meta. Aproveché el sistema de puntuación que hice para otorgar tiradas en base a los puntos. Mientras mejor era una palabra, más puntos te otorgaba que se traducían en más tiradas. El juego crecía en dificultad y era más atractivo, olvidabas que se trataba de un juego para aprender léxico.

6.1.4. Unión de componentes

Teníamos los dos componentes pero estaban separados, había que buscar la forma de unirlos. Tras un tiempo llegué a la conclusión de que el bucle sería el siguiente: crear una palabra. Asignar las tiradas de acuerdo a los puntos. Entrar al minijuego. Utilizar las tiradas. Volver a la formación de palabras y ceder el turno (en el caso del modo Player VS CPU). Era simple a la vez que entendible.

6.1.5. Mejoras e UI

Cuando estaba hecha la parte más dura del software, comencé a añadir menús de opciones, perfeccionar lo anterior, agregar sonidos que ambientaran la partida, incluir nuevas categorías, etc.

Esta etapa fue bastante tediosa, aunque gratificante, debido a que revisaba el código en busca de mejoras en referencia a la experiencia que había adquirido. He de decir que notar ese progreso de mis habilidades como programador me motivaba mucho a seguir mejorando.

6.2. Principales problemas de implementación

Como en todo proyecto aparecen problemas de implementación que nos hacen recordarlos por su dificultad y el tiempo que tuvimos que emplear en resolverlos. Es por ello que vamos a describir los momentos más complejos en los que nos hemos embarcado.

6.2.1. Soporte de múltiples tamaños de pantalla

Tal como se ha explicado anteriormente, *LibGDX* es un framework multiplataforma que nos permite utilizar el mismo código en distintos sistemas operativos sin sufrir apenas modificación. Esto es una gran ventaja, puesto que no es dependiente del sistema y nos permite trasladarlo sin problemas a otras plataformas.

No obstante, y tal como se describe en la wiki del framework, no hay una solución directa para soportar todas las resoluciones de pantalla. Así que nos preguntamos lo siguiente: Si hay tantos

dispositivos Android, ¿cómo conseguir que el videojuego se adapte a la pantalla de cada uno de ellos? He aquí el quid de la cuestión. Un asunto que me ha dado más de un dolor de cabeza. Al principio fue complicado dar con una solución, pero pronto descubrimos algo interesante que reduciría el impacto de este problema tan común en el mundo del desarrollo móvil.

Debido al alto número de solicitudes en referencia a este problema, los desarrolladores de *LibGDX* implementaron los *Viewports*. Un *Viewport* es la porción visible de un área 2D (que normalmente es más grande).

Como expliqué en la etapa de aprendizaje, opté or usar la librería *Scene2D* que se basa en dos elementos: Escenario y actores. Y un escenario está asociado a un *Viewport* por lo que pensé que como el escenario sería toda la pantalla, este tomaría el tamaño del *Viewport*. Establecimos un tamaño de 800x480 píxeles base para el juego, por ser una resolución de tamaño normal. Así que cuando se encontraba con una pantalla de mayor o menor tamaño, actualizábamos el *Viewport* para que se estrechara o extendiera si era necesario. El resultado nos agradó y las variaciones entre unas dimensiones y otras no eran tan significativas. Por lo que se usó esta idea para acabar con este dilema.

6.2.2. Sistema multi-idioma

Un atractivo del juego es que se pudieran jugar varios idiomas. Para ello nos hemos ayudado de ficheros que guardan las palabras con sus categorías para los distintos idiomas. Por lo que cualquier fichero con las palabras en formato CSV(*comma-separated values*) nos sirve para almacenar las palabras. Con tan solo añadir un nuevo fichero con las palabras del nuevo idiomas, podríamos incluir tantos idiomas como estimemos oportuno.

La estructura a nivel de código que hemos usado es un `HashMap` que almacena la información de los ficheros. Su gran manejo hace que el sistema sea fácil de implementar.

6.2.3. Conexión entre pantallas de juego

Otro problema que nos hemos encontrado es la conexión de las pantallas de juego Formación de palabras (figura 5.9) y Minijuego (figura 5.11).

La unión de ambas pantallas es crucial. El juego cambia completamente después de esto. Gracias a la clase `Game` y a la interfaz `Screen` de *LibGDX* fue posible. `Game` implementa métodos que nos permiten alternar las distintas pantallas de nuestro juego. Por lo que tan solo necesitábamos un conjunto de *Getters* y *Setters* para intercambiar la información entre ambas.

Las pantallas guardan su estado anterior gracias a los métodos de la interfaz `Screen` que actúan como el ciclo de vida de la aplicación, es decir, cuando se oculta la pantalla se llama al método `pause()` y posteriormente a `hide()` para dar paso al minijuego. Una vez que este termina seguía el procedimiento anterior. La pantalla de formación de palabras llamaba al método `resume()` y todo volvía a como estaba antes. Tan solo teníamos que evaluar la nueva información al cambiar de pantalla por si se tenían que hacer cosas distintas en el método `show()`, el cual se ejecuta al establecerse la pantalla en primer plano.

6.2.4. Comportamiento del jugador NPC

Al jugador que controla la máquina lo hemos llamado NPC (*Non-Player Character*). Implementar el comportamiento del jugador que controla el sistema también ha sido un reto interesante. Para ello hemos usado las funciones que respondían a los eventos del jugador controlado por el usuario y la clase `Timer`.

`Timer` es una clase que como su propio nombre indica en inglés es un temporizador. Controla las acciones del sistema para que parezca que las está haciendo otro usuario, ya que si lo hiciera a la velocidad del sistema, las operaciones las haría muy rápido y no daría tiempo a apreciarlas. Un objeto de la clase `Timer` programa tareas con sus métodos `scheduleTask()` que se ejecutan en el tiempo y con las condiciones que se indiquen. Estas acciones se ejecutan en un hilo paralelo al del juego. Esto es lo que ha dado lugar a más problemas, controlar ambos hilos. Con tiempo y entendiendo como funcionan los hilos, es posible coordinarlos y conseguir el resultado deseado.

6.2.5. Datos persistentes

Las aplicaciones en general guardan datos y nuestro juego necesita almacenar las estadísticas generales y de las categorías.

Para las estadísticas generales hemos usado la clase *Preferences*. Dicha clase, proporciona los métodos para guardar datos simples, como enteros, cadenas, flotantes, etc. Funcionan como una tabla, es decir, almacenan los valores tal que un hash. Por ejemplo, si queremos escribir un entero solo tenemos que usar el método `putInteger("myInteger", i)` y se crea la entrada en el fichero. Se almacenan en un fichero *XML*.

Para las estadísticas por categoría no ha sido tan sencillo, puesto que teníamos que guardar conjuntos más grandes de datos. Sin embargo, la librería de *Java* está preparada con muchos métodos de entrada/salida. En nuestro caso, guardábamos los objetos directamente mediante los métodos oportunos en ficheros. Esto ha sido completamente nuevo para mí, nunca había guardado datos más complejos que cadenas o números. Desconocía que se pudieran guardar objetos directamente y recuperarlos de forma tan efectiva.

Los ficheros se guardan en la carpeta de usuario en el caso del PC y en la tarjeta SD en el caso de Android (para ahorrar memoria interna).

6.2.6. Codificación de ficheros

Este asunto es un contratiempo derivado del apartado anterior y en relación a los caracteres especiales que mantienen tanto el Alemán como el Español. Al leer las cadenas del fichero no se usaba ninguna codificación para leerlo, por lo que generaba problemas al realizar operaciones con dichos caracteres. La solución que se tomó fue cambiar la codificación de los ficheros que contaban con dichos caracteres especiales a *UTF-8* y tenerlo en cuenta tanto en la lectura como en la escritura. Un detalle simple, pero al no estar acostumbrado a este tipo de situaciones, por la complejidad que conllevan los idiomas en cuestión, se pasa un mal momento que no se volverá a olvidar a la hora de tratar con datos de entrada y salida.

6.3. Sistema de control de versiones

En todo momento hemos estado usando el sistema de control de versiones *Git* para el apoyo en la implementación.

Siempre que se trabaja con un proyecto de cierto tamaño es bueno tener una copia de seguridad. Aún teniendo esa copia local podemos llegar a algún punto del desarrollo en que nuestro programa no funcione y tengamos que volver a un estado anterior debido a que la copia local también lo tiene. Es por esa razón que hemos usado esta potente herramienta, para cuando tengamos que recuperar una versión anterior del software.

Hemos usado *GitHub* para alojar nuestro código. Está disponible en la siguiente dirección:

https://github.com/adrianoubk/Missing_Words

6.4. Documentación del código

A lo largo del desarrollo el código se ha ido comentando internamente para las partes donde se necesitaba explicar explícitamente lo que se pretendía hacer. Con esto conseguimos que el código se pueda retomar en cualquier momento, además de ser más legible.

Adicionalmente se ha documentado el código usando *Javadoc* para hacerlo más entendible. La documentación se puede encontrar en la forja del proyecto.

Capítulo 7

Pruebas

En cualquier proyecto de desarrollo de software es más que necesario realizar una serie de pruebas para comprobar su correcto funcionamiento.

Para el diseño de los casos de prueba hemos distinguido entre los elementos que se pueden probar individualmente frente a los que necesitan hacerlo de manera integrada. Cuando se implementaba un módulo nuevo era exhaustivamente testeado, controlando que se cubrieran todas las opciones disponibles. Algunas implementaciones no recibían un caso especial de prueba por su simplez, por lo que era mejor esperarse a su integración.

En definitiva, la organización de los casos de prueba fue la siguiente:

1. Se realizaban las pruebas cada vez que se implementaba un módulo nuevo. Al principio se probaban individualmente, por no haber otros.
2. Una vez que su funcionamiento era correcto, se pasaba a cohesionar los módulos implementados y efectuar su prueba conjuntamente.
3. Cuando había una cantidad considerable de módulos y las posibilidades eran mayores, se lanzaba una nueva versión para que la probaran otros usuarios además del desarrollador.

7.1. Pruebas unitarias

Se han ido haciendo pruebas unitarias constantemente durante todo el desarrollo, es decir, cada vez que se creaban elementos que precisaban la atención necesaria. Esto garantizaba que no nos encontráramos con imprevistos por no haberlo probado totalmente.

Esto incluía cualquier clase que superara la complejidad estimada. Por ejemplo, la clase que controla la barra de tiempo, donde hay que manejar varios factores o la interacción entre el usuario y los eventos del sistema.

Este mecanismo nos ha permitido construir una base sólida, encontrar los errores de una manera más precisa e incluso ayudar a entender mejor el código.

7.2. Pruebas de integración

Alcanzado el punto en el que era necesaria una prueba que pusiera en acción varios módulos que habían sido testeados particularmente, es donde tienen lugar las pruebas de integración.

Esta fase conlleva por ejemplo, comprobar que todas las pantallas se coordinaban y que se podían enlazar unas con otras. La unión de las pantallas de formación de palabras y minijuego es fundamental, ya que conforman el juego como tal. Esta última ha consumido una gran cantidad de tiempo, porque conseguir que ambas pantallas se conjuntaran no ha sido fácil. A través de depurar el código, explorar todos los caminos posibles y con el esfuerzo empleado, se consiguen resultados satisfactorios.

7.3. Pruebas funcionales

En estas pruebas, comprobamos que el sistema se ajusta al análisis y diseño realizado. Como parte de ella está el correcto funcionamiento de la interfaz gráfica. Una vez implementadas las pantallas y la navegación entre menús, solo hay que verificar que funcionan adecuadamente. De esta forma comprobamos que la interacción del usuario es completa y cumple el modelo de casos de uso (figura 4.1).

7.4. Pruebas no funcionales

En esta prueba se ha intentado medir la eficiencia del software. Para ello hemos probado el software bajo distintos sistemas operativos (que tuvieran *Java* instalado en el caso de Escritorio) como Windows (XP, 7 y 8) y Ubuntu (12.04 y 14.04) para plataformas de PC. En el caso de Android hemos probado el producto en gran cantidad de dispositivos, tanto físicos como virtuales, que tuvieran instaladas desde la versión 2.3.x hasta la versión 4.4.2.

Hemos podido comprobar que en escritorio funciona perfectamente. En Android también ha tenido un resultado satisfactorio. Si bien es cierto, que en dispositivos que tienen menos recursos se ve afectado un poco el rendimiento de la aplicación, pero nada comparado con imposibilitar su uso. Esto por supuesto, es en dispositivos bastante antiguos que no son ni comparables con los que hay actualmente en el mercado.

Esta percepción es interesante, puesto que hemos tenido en cuenta que los terminales más antiguos y limitados puedan instalar el juego.

7.5. Pruebas de aceptación

Finalmente se han llevado a cabo pruebas de aceptación, las cuales han consistido en distribuir el producto entre varias personas para averiguar posibles fallos de la aplicación además de recibir comentarios sobre mejoras o funcionalidades nuevas.

Estas pruebas de versiones no solo han sido completadas por los tutores del proyecto, sino también por amigos o familiares. Como resultado, hemos recibido numerosas opiniones que nos han permitido seguir mejorando. La mayoría de las funcionalidades sugeridas se han implementado. Asimismo, nos ha sido de gran ayuda en las pruebas no funcionales, ya que se ha podido testear bajo distintos móviles con distintas versiones y características.

Hay que mencionar que se han centrado sobre todo en probar la plataforma móvil más que la de PC, por una razón obvia, la de tener un mayor número de terminales de prueba accesibles.

Evidentemente, han intentado medir la experiencia del usuario y la jugabilidad. En ocasiones, hemos cambiado el diseño de ciertas partes de la aplicación. Por ejemplo, la disposición de la pantalla de formación de palabras, donde los botones estaban muy juntos y no se podían pulsar con facilidad fue reportado por los usuarios y se decidió cambiar el diseño de la misma, colocando estos en los laterales.

Capítulo 8

Conclusiones

En este capítulo realizaremos una reflexión de lo que ha supuesto la creación de *MissingWords*.

8.1. Experiencia personal

MissingWords ha sido mi primer proyecto y es evidente que no ha sido una tarea sencilla. He recorrido un largo camino, desde el aprendizaje de herramientas hasta la última línea de código escrita.

En primer lugar, decir que nunca me había enfrentado a un proyecto de tal tamaño y que no es nada comparable a los pequeños proyectos que se hicieron en los años anteriores. Obviamente, no pretendo quitarle valor a los conocimientos adquiridos a lo largo de la titulación, es más, han establecido los cimientos de este proyecto en cierto sentido. Y es que sin todos estos años, probablemente no se hubiera alcanzado este objetivo.

Seguidamente, he de destacar el estudio de todas las herramientas así como librerías que han hecho posible el software.

Empezando por las herramientas gráficas, he de admitir que ha sido una parte más que tediosa que unida a mis nefastas habilidades como “artista” lo han hecho peor. Después de todo, me he sorprendido a mí mismo y creo que el resultado aunque no profesional, es decente para la vista. Y Por supuesto me ha servido para reconocer el trabajo de la persona que se encarga de hacer esto. Desde luego es admirable.

En cuanto a *LibGDX*, es un proyecto del que los desarrolladores tienen que estar más que orgullosos. A mí por lo menos me ha sido de gran valor, he aprendido los fundamentos de un videojuego a la vez que he mejorado mis habilidades como programador. Me ha dado la oportunidad de conocer como se construye una aplicación gráfica e incluso de hacer mi primer juego.

Y así con muchas otras que podría describir. El caso es que cada una de ellas ha supuesto un esfuerzo que se ha convertido en saber y como *el saber no ocupa lugar*, o eso dicen (tiempo desde luego sí), merece la pena que sea mencionado, pues es al fin y al cabo lo que te llevas de todo el viaje.

Por último, pero no menos importante, he aprendido lo imprescindible que es la planificación y la organización del trabajo. El controlar el tiempo que tienes para cada tarea, los pasos que tienes que dar, probar cada unidad independiente antes de probarlo íntegramente y todos esos procesos que constituyen la ingeniería del software. Todos, son necesarios.

Sin extenderme más, espero que este software sea además un complemento en el aprendizaje de idiomas y que se use en el ámbito de la docencia (y sino, me quedo con la experiencia que ha implicado), porque si hay algo que todos sabemos es que cuando algo es divertido, seguro que se ha aprendido.

8.2. Mejoras futuras

Como en todo desarrollo del software, el ciclo nunca termina y siempre se pueden incluir mejoras. Estas podrían ser las siguientes:

- **Modo multijugador:** En vez de que el jugador se enfrente a la máquina, se podría hacer entre dos jugadores tanto de forma síncrona como asíncrona.
- **Logros:** Incluir logros que motiven a los jugadores a conseguirlos como crear 100 palabras, ganar 10 juegos seguidos, etc.
- **Mejora de gráficos:** Mejorar la interfaz de usuario, mediante elementos gráficos de mayor calidad para hacer más agradable la experiencia del usuario.
- **Estadísticas:** Ampliar el conjunto de estadísticas así como buscar una forma mejor de mantenerlos mediante una base de datos en un servidor.
- **Más minijuegos:** Añadir otros minijuegos puede ser una propuesta interesante, que haría las partidas aún más espontáneas y dinámicas. Aumentaría mucho su re-jugabilidad.

Herramientas utilizadas

Framework

LibGDX es un framework multiplataforma para el desarrollo de videojuegos 2D y 3D multiplataforma. Se escogió por su gran adaptación y documentación.



Herramientas: Logo de LibGDX

Lenguaje de programación

El lenguaje de programación que hemos usado es *Java* debido a que el framework lo usa.



Herramientas: Logo de Java

Herramientas gráficas

Photoshop y Gimp

Herramientas para el diseño y retoque gráfico. Con ellas hemos creado y editado los gráficos del juego.



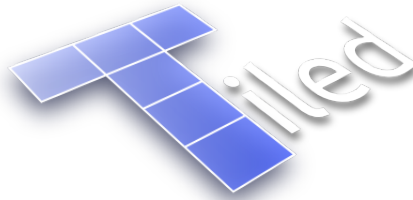
Herramientas: Logo de Photoshop



Herramientas: Logo de Gimp

Tiled

Tiled es un editor de mapas organizado en Tiles tal como su propio nombre indica. Es muy usado para crear mapas de baldosas en el mundo de los videojuegos. Con este software hemos creado el tablero del minijuego.

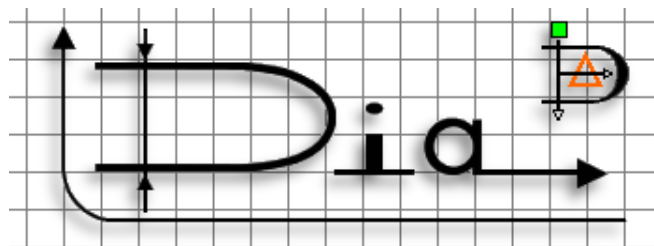


Herramientas: Logo de Tiled

Elaboración de diagramas

Para realizar los diagramas que se han usado en esta memoria hemos usado *Dia*.

Dia es un programa para la creación general de diagramas. Se pueden dibujar diferentes tipos de diagramas entre ellos diagramas de entidad-relación, diagrama de casos de uso, etc.



Herramientas: Logo de Dia

Documentación del código

Para la documentación hemos usado *Javadoc*, por ser el estándar para documentar clases en *Java* y por su fácil y clara generación. La documentación se crea en formato *HTML*

Sistema de control de versiones

Para tener un seguimiento de las versiones del proyecto así como recuperar una versión anterior del programa en caso de problemas hemos usado *Git*. Todo nuestro código está alojado en *GitHub* que utiliza este sistema.



Herramientas: Logo de Git

Eclipse

Eclipse es un software que proporciona un conjunto de herramientas de programación para la elaboración de proyectos.



Herramientas: Logo de Eclipse

El por qué de usar *Eclipse* es porque era necesario cuando estaba haciendo el curso de *Android* y me acostumbré a su uso. *LibGDX* también está muy enfocado para importar proyectos a este programa.

Redacción de la memoria

Para la completa redacción de esta memoria hemos usado \LaTeX . es un sistema de composición de textos, orientado a la creación de documentos específicos y técnicos que incluyen entre otras cosas, formulas matemáticas.



Herramientas: Logo de Latex

Manual de instalación

Instalación

En primer lugar, para ejecutar *MissingWords* tenemos que descargar el archivo necesario (PC o Android) desde la página siguiente:

https://github.com/adrianoubk/Missing_Words/releases

Nos aparecerá una página como esta:



Descarga *MissingWords*

PC

Para usar *MissingWords* en nuestro PC es imprescindible tener instalado en nuestro ordenador *Java 7* o superior. Si no lo tenemos, podemos descargarlo de la página oficial:

<https://www.java.com/es/download/>

Una vez tengamos *Java*, debemos descargar el archivo **MissingWords.jar** que se muestra en la imagen anterior. Para empezar a jugar tan solo hay que ejecutarlo.

Android

Igualmente para Android, tenemos que descargar el fichero **MissingWords.apk** y ejecutarlo para empezar a instalar la aplicación y empezar a jugar.

Debemos asegurarnos de que tengamos habilitado el permiso para instalar aplicaciones de fuentes desconocidas en los ajustes del teléfono.

Manual de usuario

A continuación facilitamos el manual de *MissingWords v1.0*. Con este manual pretendemos ampliar el breve resumen del juego que se encuentra en la pantalla de instrucciones (figura 5.8) sobre como jugar.

MissingWords

MissingWords es un juego educativo para aprender vocabulario de idiomas, con un extra de diversión. Nuestro objetivo es superar un minijuego mediante la formación de palabras.

Todo el código del juego está disponible en la página de la forja para su descarga, modificación y uso bajo licencia *GPLv3* incluida al final de este documento.

https://github.com/adrianoubk/Missing_Words

Información del juego

Modos de juego

El juego tiene disponibles dos modos de juego: Jugador VS CPU y Un jugador.

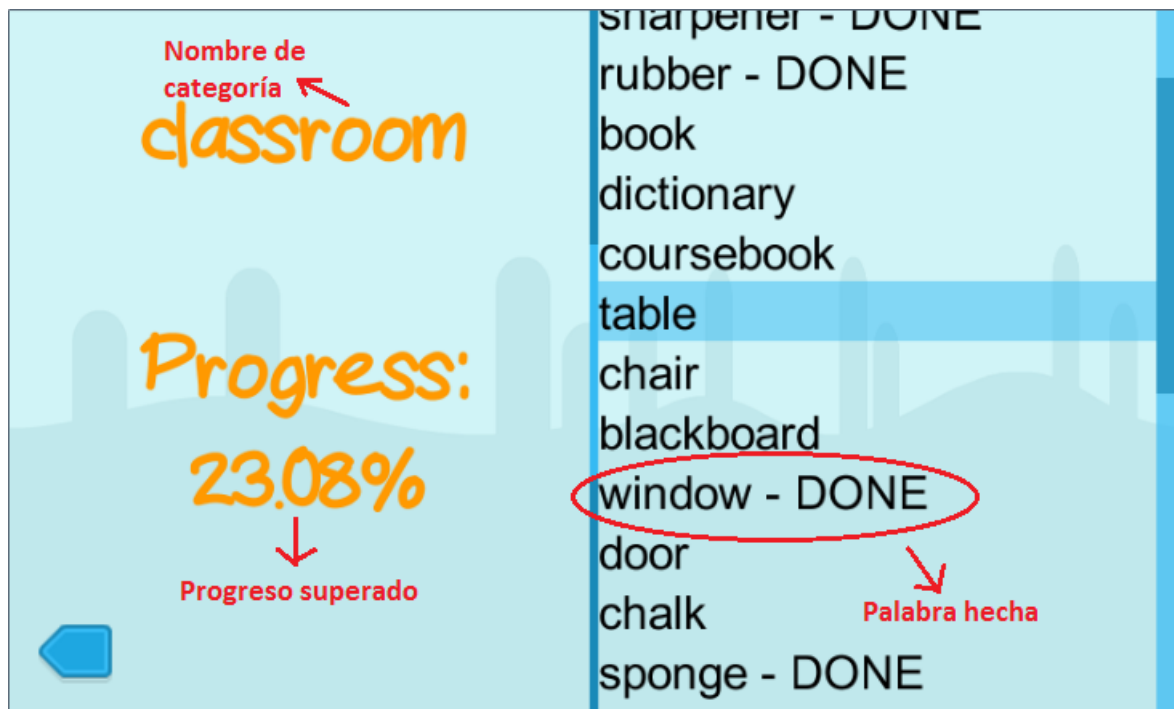
En el primero el jugador se enfrenta a la máquina para ganar el minijuego. En el segundo solo juega el jugador con el mismo objetivo. En este último, solo aparecería el personaje del jugador en el minijuego.

Estadísticas

Dentro de la pantalla estadísticas (figura 5.5) tenemos las estadísticas generales y por categoría.

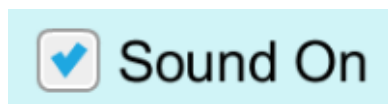
Las primeras como su propio nombre indica nos muestra información genérica del juego, es decir, partidas ganadas, partidas perdidas, pistas usadas, mejor palabra (más puntos), etc.

Las estadísticas por categoría en cambio, nos muestra el progreso que llevamos en una categoría, las palabras que hay en dicha categoría y las que hemos hecho. En la siguiente imagen se aprecia mucho mejor:

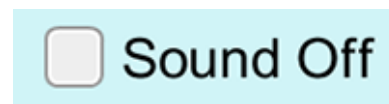


Ajustes

En la pantalla de ajustes podemos activar o desactivar el sonido de la aplicación, además de ver los créditos de la misma (figura 5.7). Los ajustes no se guardan al salir de la aplicación.



Sonido activado



Sonido desactivado



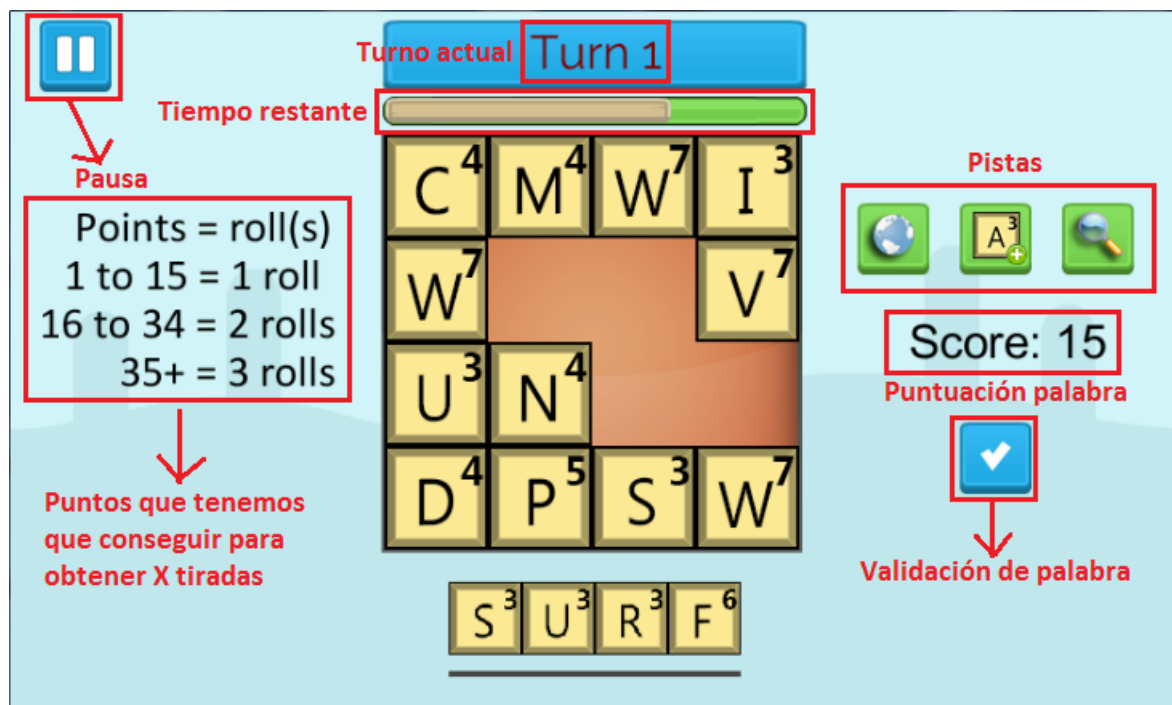
Idioma seleccionado

Jugar

A continuación vamos a detallar como se juega a *Missing Words*. Para empezar a jugar tenemos que escoger uno de los dos modos disponibles desde el menú principal. Seguidamente se nos pedirá una categoría para jugar (figura 5.4) y entraremos al juego.

Pantalla de juego: Formación de palabras

Cuando entramos a la partida lo primero que vemos es un mensaje que nos indica que es nuestro turno. Primero juega el jugador y luego la máquina. En esta pantalla tenemos que ir seleccionando las letras con el ratón o dedo, dependiendo de la plataforma donde estemos jugando, para ir bajando las letras y formar una palabra. El proceso inverso devuelve la ficha a la caja de letras. A continuación una imagen resumen de los elementos:



Formacion de palabras - Elementos

Descripción de elementos

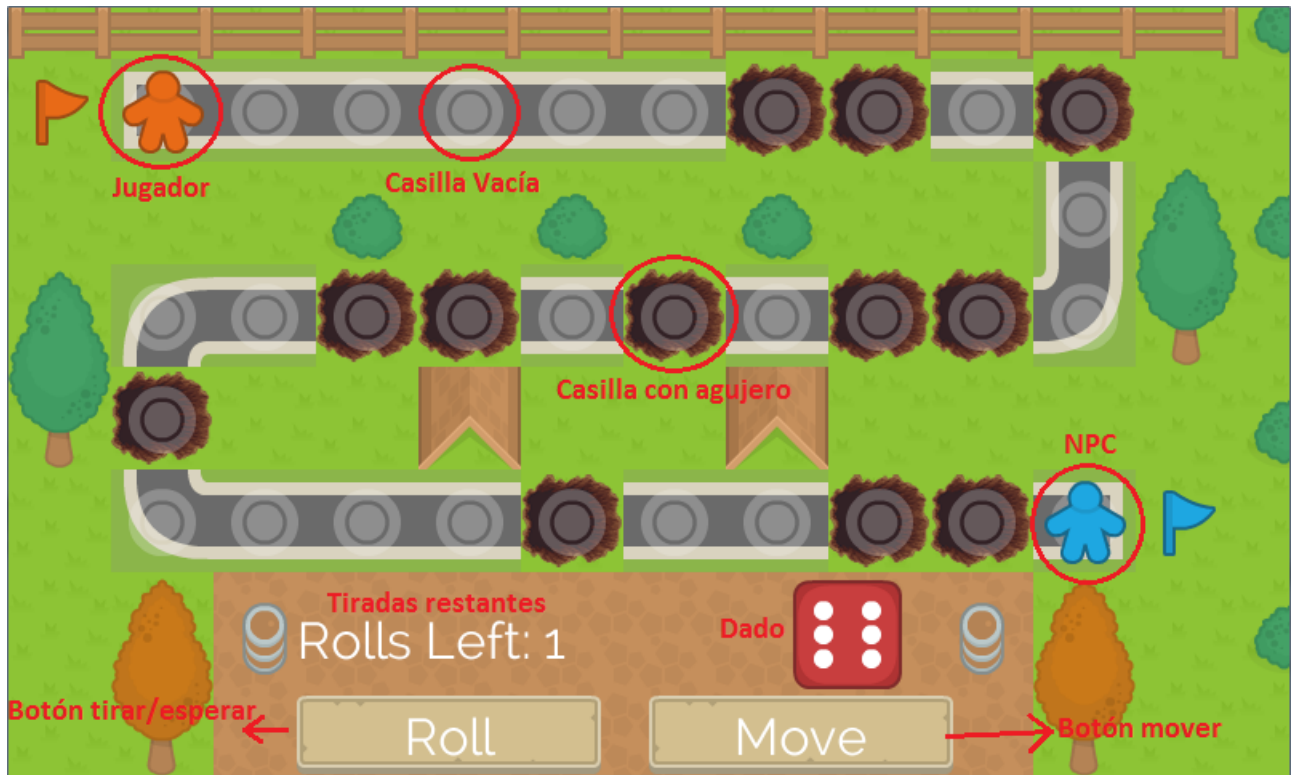
Vamos a describir los elementos de la imagen:

- ◇ *Botón de pausa*: Pausa el juego en cualquier momento (figura 5.10). Al entrar el menú de pausa se para el tiempo y se oculta la caja de letras. Ahí tenemos dos opciones: reanudarlo o salir. Si lo reanudamos se vuelve al juego. Si por el contrario salimos, volvemos al menú principal **perdiendo** los progresos.

- ◇ *Información de tiradas*: Muestra los puntos que tenemos que conseguir para obtener X tiradas. Por ejemplo, una palabra de 16 a 34 puntos nos otorgaría dos tiradas.
- ◇ *Tiempo restante*: Es el tiempo que nos queda para formar una palabra. Si este acaba, se le cede el turno a la máquina.
- ◇ *Turno actual*: Indica el turno que estamos jugando en este momento.
- ◇ *Pistas*: Sólo se pueden utilizar dos pistas por turno. Si usamos dos, nos restarían las tres tiradas, pero el mínimo siempre es una al formar una palabra. Hay tres tipos de pista:
 - Traducción: Representada con un globo terráqueo, nos traduce una de las palabras que haya en el tablero al español. Nos resta dos tiradas.
 - Letra adicional: Representada con una ficha, nos da la primera letra de una de las palabras del tablero. Nos resta una tirada.
 - Longitud: Representada con una lupa, nos dice la longitud de una de las palabras del tablero. Nos resta una tirada.
- ◇ *Puntuación de palabra*: puntuación de la palabra que estamos creando actualmente.
- ◇ *Botón de validación*: Verifica que la palabra existe y pertenece a la categoría. Si la palabra es correcta nos lleva al minijuego, en caso contrario nos lo notifica.

Pantalla de juego: Minijuego

Ahora es el momento de agotar las tiradas que conseguimos en la pantalla anterior. El objetivo es llegar al otro extremo del tablero con las tiradas y sin caer en los agujeros. Veamos la pantalla del minijuego y sus elementos:



Minijuego - Elementos

Descripción de elementos

Los elementos son los siguientes:

- ◇ *Jugador*: Es la ficha que representa al jugador en el tablero.
- ◇ *NPC*: Es la ficha que representa al jugador controlado por la máquina en el tablero.
- ◇ *Casilla vacía*: Casilla que conforma el camino.
- ◇ *Casilla con agujero*: Casilla que representa un agujero en el camino, caer en él nos haría volver al principio.
- ◇ *Tiradas restantes*: Son las tiradas que nos quedan disponibles para jugar.
- ◇ *Dado*: Representa la tirada actual que hemos conseguido.
- ◇ *Botón tirar/esperar*: Botón con el que se tira el dado. Cambia a esperar cuando no quedan tiradas restantes y no se ha movido. Lo explicaremos en el siguiente apartado.
- ◇ *Botón mover*: Mueve al jugador tantas casillas como indica el dado.

Mecánica del minijuego

La mecánica del minijuego es sencilla. Con las tiradas que tenemos lanzamos el dado y movemos a nuestro personaje hasta llegar al otro extremo procurando no caer en los hoyos.

Para ello tiraremos pulsando el botón *Roll*. Ahora que sabemos cual ha sido nuestra tirada, tenemos que decidir si movernos o no. Aquí nos encontramos con 3 casos distintos:

- **Tirar de nuevo:** Tenemos tiradas restantes y no nos agrada el resultado. Podemos tirar de nuevo para obtener una nueva tirada.
- **Movernos:** Mover nuestro personaje tantas casillas como indica nuestra tirada.
- **Esperar:** Podemos pasar el turno si hemos movido al menos una de nuestras tiradas.

Los dos primeros casos no tienen problema. Sin embargo, **Esperar** es un caso más especial. Podemos pensar que esto es una ventaja, pero no es tan fácil como parece. Y es que para usar esta opción tenemos que haber movido al menos una de nuestras tiradas. ¿Qué significa esto? Que si esta restricción no existiera, siempre esperaríamos hasta sacar en la tirada un resultado a nuestro favor. Por lo que si solo tienes una tirada, debes mover obligatoriamente. Si intentas esperar sin haber movido antes, el sistema te mostrará el mensaje: “Debes mover al menos una vez”. En la siguiente imagen lo apreciaremos mejor.



Minijuego - Función esperar

Fin del juego

El juego termina cuando uno de los jugadores (o el jugador en el modo solitario) alcanza la meta. Se le mostrará entonces la pantalla de victoria (figura [5.12](#)).

Manual del profesor

Este manual se ha hecho con el objetivo de que el profesor pueda añadir nuevo vocabulario a las categorías existentes del juego, así como modificar los umbrales de puntos a la hora de otorgar tiradas para el minijuego, haciendo posible que se puedan personalizar los contenidos del mismo.

Introducción

MissingWords es un juego para aprender vocabulario de idiomas basado en categorías. Estas comprenden los temas más relevantes que se estudian en una lengua extranjera al principio de su aprendizaje.

Como estos contenidos pueden ser muy extensos, se otorga la posibilidad de ampliar el vocabulario de las categorías que contiene el juego de una manera muy sencilla.

Adicionalmente, se pueden modificar los umbrales de puntos, que delimitan como se obtienen las tiradas.

Categorías del juego

Hay un total de 11 categorías en el juego. Son las siguientes:

- Días (*Days*).
- Meses (*Months*).
- Preguntas con W (*W-Questions*).
- Colores (*Colours*).
- Tamaño (*Size*).
- Aula (*Classroom*).
- Partes del cuerpo (*Body-parts*).

- Sentimientos (*Feelings*).
- Universidad (*University*).
- Ciudad (*City*).
- Tiempo libre (*Freetime*).

¿Cómo se añade nuevo vocabulario al juego?

Para añadir nuevo vocabulario al juego primero tenemos que saber donde se guardan los datos en *MissingWords* y cual es su estructura de directorios.

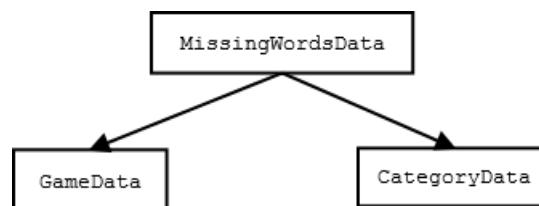
Datos

MissingWords guarda los datos en localizaciones distintas según estemos ejecutando el juego bajo PC o Android.

En PC, podemos localizar el directorio en nuestra carpeta de usuario.

En Android, lo podemos encontrar en la raíz de nuestra tarjeta SD.

Los datos del juego se guardan en el directorio **MissingWordsData**. Este a su vez tiene dos subdirectorios **GameData** y **CategoryData**.



Organización de datos en *MissingWords*

GameData: Es el directorio que nos interesa. Aquí es donde añadiremos el vocabulario con un formato que explicaremos a continuación.

CategoryData: Directorio que guarda las estadísticas de las categorías. **No** se debe tocar.

Subdirectorio *GameData*

Dentro de este directorio nos encontramos con los siguientes ficheros:

Ficheros de vocabulario:

- `vocabulary-english.txt`
- `vocabulary-german.txt`

Ficheros de traducción (para la pista traducción):

- english-spanish.txt
- german-spanish.txt

Ficheros de umbrales de puntos:

- score-limits-english.txt
- score-limits-german.txt

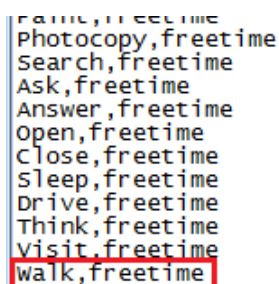
Vocabulario nuevo

Cambiar el vocabulario es tan sencillo como añadir una línea al fichero de vocabulario del idioma que queramos **respetando la primera línea del fichero que está vacía y no se debe escribir ahí**. Para ello abriremos el fichero e introduciremos una línea con el siguiente formato:

PalabraVocabulario,categoría

Hay que tener en cuenta mayúsculas, minúsculas y espacios. **PalabraVocabulario** debe empezar con mayúsculas, la categoría con minúscula y no haber espacios (solo la coma). Evidentemente la categoría deber ser una de las listadas anteriormente y en **inglés** siempre (aunque cambiemos el fichero alemán). Por ejemplo supongamos que queremos añadir una palabra nueva al vocabulario inglés de la categoría “Freetime”. Procederíamos de la siguiente forma:

Walk,freetime



```
Paint,freetime
Photocopy,freetime
Search,freetime
Ask,freetime
Answer,freetime
Open,freetime
Close,freetime
Sleep,freetime
Drive,freetime
Think,freetime
Visit,freetime
walk,freetime
```

Organización de datos en *MissingWords*

Por supuesto, se puede insertar la línea donde se quiera, pero se recomienda justo después de la última palabra, ya que las estadísticas por categoría siguen el orden del fichero. O cambiar este a juicio propio.

Ahora podemos poner la traducción de la palabra en el fichero correspondiente de traducción con el mismo formato, pero sustituyendo la categoría por su traducción. Siguiendo nuestro ejemplo, en el fichero english-spanish.txt:

Walk,Pasear

La traducción se puede poner tanto en mayúsculas (en el vocabulario incluido se pone en mayúsculas) como en minúsculas e igualmente no tiene por qué seguir un orden.

También, se pueden **borrar** las palabras si hemos cometido un error o queremos quitar algunas de las que trae el juego por defecto. Para llevarlo a cabo, simplemente se borra la línea que no queramos de ambos ficheros.

Ajustar umbrales de puntos

Para ello utilizaremos los ficheros restantes. Estos ficheros solo tienen dos líneas que son las siguientes:

max X
min Y

El primero indica el valor máximo y el segundo el mínimo, tan solo tenemos que cambiarlos, respetando el espacio pertinente. Por ejemplo,

max 35
min 15

Esto ajustaría los umbrales en la pantalla de formación de palabras de la siguiente forma:

1 to 15 = 1 roll
16 to 34 = 2 rolls
35+ = 3 rolls

Estos son los valores por defecto, tal como podemos observar en figura [5.9](#).

Ya tengo mis ficheros modificados, ¿y ahora qué?

Si queremos que los alumnos tengan el vocabulario nuevo, tan solo tenemos que enviarles el fichero y que lo copien en el directorio especificado. Esto **no** hará que pierdan su progreso en las categorías alteradas, solo añadirá las nuevas palabras.

Bibliografía

- [1] Wiki de LibGDX. <https://github.com/libgdx/libgdx/wiki>
- [2] Documentación de LibGDX. <http://libgdx.badlogicgames.com/nightlies/docs/api/>
- [3] Documentación de Java. <https://docs.oracle.com/javase/7/docs/api/>
- [4] Head First Java. Kathy Sierra & Bert Bates. ISBN-13: 978-0596009205
- [5] Beginning Android Games Second Edition. Mario Zechner & Robert Green. ISBN-13: 978-14302467705
- [6] Libgdx Cross-platform Game Development Cookbook. David Saltarez Márquez & Alberto Cejas Sánchez. ISBN-13: 978-1783287291
- [7] Extracto del libro Games and Learning Alliance: Learning Analytics and Educational Games. Ángel Serrano-Lagunaa, Javier Torrente, Borja Maneroa, Ángel del Blancoa, Blanca Borro-Escribanoa, Iván Martínez-Ortiza, Manuel Freire, and Baltasar Fernández-Manjón. ISBN-13: 978-3319121567
- [8] Ayuda oficial de Photoshop. <http://helpx.adobe.com/es/photoshop/topics.html>
- [9] Documentación oficial de Gimp. <http://docs.gimp.org/2.8/es/>
- [10] Wiki de Tiled. <https://github.com/bjorn/tiled/wiki>
- [11] Documentación de Latex. <http://en.wikibooks.org/wiki/LaTeX>
- [12] Preguntas y respuestas. <http://stackoverflow.com/> & <http://www.badlogicgames.com/forum/>

GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.

<<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document “free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “**Document**”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “**you**”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “**Modified Version**” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “**Secondary Section**” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “**Invariant Sections**” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “**Cover Texts**” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “**Transparent**” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “**Opaque**”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “**Title Page**” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text. The “**publisher**” means any person or entity that distributes copies of the Document to the public.

A section “**Entitled XYZ**” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “**Acknowledgements**”, “**Dedications**”, “**Endorsements**”, or “**History**”.) To “**Preserve the Title**” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included

by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

- M. Delete any section Entitled “Endorsements”. Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled “Endorsements” or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section Entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements”.

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>. Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

“Massive Multiauthor Collaboration Site” (or “MMC Site”) means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A “Massive Multiauthor Collaboration” (or “MMC”) contained in the site means any set of copyrightable works thus published on the MMC site.

“CC-BY-SA” means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

“Incorporate” means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is “eligible for relicensing” if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright © YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with ... Texts.” line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

GPL General Public License

Version 3, 29 June 2007
Copyright © 2007 Free Software Foundation, Inc.

`<http://fsf.org/>`

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

PREAMBLE

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program—to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to

the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The “source code” for a work means the preferred form of the work for making modifications to it. “Object code” means any non-source form of a work.

A “Standard Interface” means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The “System Libraries” of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A “Major Component”, in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The “Corresponding Source” for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work’s System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright.

Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

- d)* If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an “aggregate” if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation’s users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

- a)* Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
- b)* Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.
- c)* Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.
- d)* Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.
- e)* Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A “User Product” is either (1) a “consumer product”, which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, “normally used” refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

“Installation Information” for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

“Additional permissions” are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but

the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a)* Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b)* Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c)* Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d)* Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e)* Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f)* Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered “further restrictions” within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically

terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An “entity transaction” is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party’s predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A “contributor” is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor’s “contributor version”.

A contributor’s “essential patent claims” are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, “control” includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor’s essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a “patent license” is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To “grant” such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. “Knowingly relying” means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient’s use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is “discriminatory” if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License “or any later version” applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF

MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the “copyright” line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <textyear> <name of author>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
<program> Copyright (C) <year> <name of author>
```

```
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it  
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w` and `show c` should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an “about box”.

You should also get your employer (if you work as a programmer) or school, if any, to sign a “copyright disclaimer” for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <http://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <http://www.gnu.org/philosophy/why-not-lgpl.html>.